PB82-254756

Users Manual and Program Guide for the Consumer Product Noise Impact Model

Bolt Beranek and Newman, Inc.
Cambridge, MA

Prepared for

Environmental Protection Agency
Washington, DC

Dec 81

U.S. Department of Commerce
National Technical Information Service
NTIS

EPA 550/9-81-250

PB82-254756

USERS MANUAL AND PROGRAM GUIDE FOR THE CONSUMER PRODUCT NOISE IMPACT MODEL

December 1981

U. S. Environmental Protection Agency
Washington, D.C. 20460

REPRODUCED BY
NATIONAL TECHNICAL
INFORMATION SERVICE
U.S. DEPARTMENT OF COMMERCE
SPRINGFIELD, VA. 22161

ATTENTION

PORTIONS OF THIS REPORT ARE NOT LEGIBLE. HOWEVER, IT IS THE BEST REPRODUCTION AVAILABLE FROM THE COPY SENT TO NTIS.

USERS MANUAL AND PROGRAM GUIDE FOR THE CONSUMER PRODUCT NOISE IMPACT MODEL

DECEMBER 1981

OFFICE OF NOISE ABATEMENT AND CONTROL
U.S. ENVIRONMENTAL PROTECTION AGENCY
WASHINGTON, D.C. 20460

This report has been approved for general availability. The contents of this report reflect the views of the contractor, who is responsible for the facts and the accuracy of the data presented herein, and do not necessarily reflect the official views or policy of EPA. This report does not constitute a standard, specification, or regulation.

Permission is granted to reproduce this material without further clearance.

50272-101

REPORT DOCUMENTATION PAGE

1. REPORT NO. EPA - 550/9-81-250

2.

3. Recipient's Accession No. PB82 254756

4. Title and Subtitle
Users Manual and Program Guide for the
Consumer Product Noise Impact Model

5. Report Date October 1980

6.

7. Author(s) R. L. Bronsdon and C. E. Waldman

8. Performing Organization Rept. No. BBN 4510

9. Performing Organization Name and Address
Bolt Beranek and Newman, Inc.
50 Moulton Street
Cambridge, MA 02238

10. Project/Task/Work Unit No.

11. Contract(C) or Grant(G) No.
(C) EPA 68-01-3988
(G)

12. Sponsoring Organization Name and Address
Office of Noise Abatement and Control
U.S. Environmental Protection Agency
Washington, D.C. 20460

13. Type of Report & Period Covered Final

14.

15. Supplementary Notes
This one volume report containing both a users manual and a programers guide, as well as a computer tape are available from NTIS.

16. Abstract (Limit: 200 words)

The Consumer Product Noise Impact Model (CPNIM) was written to enable the Environmental Protection Agency's (EPA) Office of Noise Abatement and Control to assess the impact of noise from consumer products on the public. The original model was designed by the National Bureau of Standards (NBS) in 1978 to set the basic calculation procedures.

Two versions of this program have been provided. The first, and easiest to operate, is interactive in nature and is designed to be operated by a person possessing virtually no computer skills. It operates in the "Time Sharing" mode as opposed to the "Batch" mode and is, as such, more expensive to run, but the extent of the user's knowledge of the programs workings is limited to how to start the program.

The second version is designed to operate in the "Batch" mode. To operate this version the user must be capable of using the terminal to build files for input data and submitting a batch job for execution. All steps are carefully explained in this manual but some knowledge of computing systems is helpful. Because of the "Batch" operation, this version is much less expensive to run, but the turn around time will be longer.

17. Document Analysis a. Descriptors
Consumer Products
Noise Impact
Computer Model

b. Identifiers/Open-Ended Terms

c. COSATI Field/Group

i.a

18. Availability Statement Release Unlimited

19. Security Class (This Report) Unclassified

20. Security Class (This Page) Unclassified

21. No. of Pages

22. Price

(See ANSI-Z39.18) See Instructions on Reverse OPTIONAL FORM 272 (4-77) (Formerly NTIS-35) Department of Commerce

FOREWORD

The user is assumed to be familiar with the follwoing documents:

- National Computer Center - IBM System, "NCC - IBM WYLBUR Guide"
- U.S. Environmental Protection Agency, "NCC - IBM User's Guide"
- IBM, "OS/VS2 TSO Command Language Reference Manual" GC28-0646-4.

This user's manual and program guide describes the consumer product noise impact model as it existed on the NCC computer system on July, 1981.

TABLE OF CONTENTS

page

1. INTRODUCTION 1
    1.1 Purpose 1
    1.2 User Profile 1
2. SYSTEM SUMMARY 3
    2.1 Overview 3
    2.2 Operational Environment 12
    2.3 System Configuration 15
3. SYSTEM OPERATION 16
    3.1 Common System Initiation Procedures 16
    3.2 Description of the Interactive Program 18
    3.3 Description of Batch Program 35
4. LOW LEVEL PROGRAM DESCRIPTION 53

APPENDIX A: PROGRAM REFERENCES A-1

APPENDIX B: GLOSSARY OF VARIABLES B-1
    B.1 Product Specific Variables B-1
    B.2 Fundamental Variables B-4
    B.3 Input Parameter Codes B-5

APPENDIX C: PROMPT FILES, HELP FILES, ERROR FILES . . . C-1

APPENDIX D: CODE FILE D-1

APPENDIX E: FUNDAMENTAL DATA SET E-1

APPENDIX F: SAMPLE OPERATIONAL PROCEDURES F-1

APPENDIX G: PROBLEMS IN EXECUTION. G-1

1. INTRODUCTION

1.1 Purpose

This computer model has been written to enable the Environmental Protection Agency's (EPA) Office of Noise Abatement and Control to assess the impact of noise from consumer products on the public. Such an assessment is necessary in determining which products in an increasingly noisy home environment need to be evaluated pursuant to a Section 6 regulatory, or Section 8 labeling, type of effort in the fulfillment of EPA's charter as set forth in the Noise Control Act of 1972. The original model was designed by the National Bureau of Standards (NBS) in 1978 to set the basic calculation procedures. In 1980 Bolt, Beranek and Newman Inc. (BBN), under EPA contract, developed a user interface and adapted the model to EPA's National Computer Center (NCC) system.

The severity of individual noise exposures is estimated by noise emission data and data on noise isolation between the source and various listener locations. The extent of the noise exposures is estimated from data on population groupings by number of people, by amount of time, by type of dwelling, by room in dwelling; further, data on number, time of day, and length of time of product utilization must be used in estimating extent of noise exposures. Finally, a procedure is needed for combining estimates of severity and extent into a single estimate of total population impact, and this is accomplished by means of weighting functions. It can be seen that the specific variables listed in Appendix B fit into the rather general categories given here. Definitions of the various weighting functions used can be found on pages 32-35 and pages 47-50.

1.2 User Profile

Two versions of this program have been provided. The first, and easiest to operate, is interactive in nature and is designed to be operated by a person possessing virtually no computer skills. This allows inexperienced personnel to process data and obtain results without extensive familiarization time. It operates in the "Time Sharing" mode as opposed to the "Batch" mode and is, as such, more expensive to run, but the extent of the user's knowledge of the programs workings is limited to how to start the program. After that, the program asks for all inputs and even answers questions about the meaning of input variables.

The second version is designed to operate in the "Batch" mode. To operate this version the user must be capable of using the terminal to build files for input data and submitting a batch job for execution. All steps are carefully explained in this manual but some knowledge of computing systems is helpful. Because of the "Batch" operation, this version is much less expensive to run, but the turn around time will be longer.

2. SYSTEM SUMMARY

2.1 Overview

This subsection consists mainly of a description of the flow diagram presented at the end of the subsection. While a major portion of this system code is designed to produce a user interface, the most important sections deal with the calculations which lead to the output values. As such, the system description will be on two levels, a very high level for user interface functions, and a lower level for impact calculations.

2.1.1 Overview of interactive program

The interactive program structure is based on a modular approach wherein most individual tasks are performed in separate subroutines. The flow discussed in this section is very high level, that is not very specific; a more detailed discussion can be found in Section 4 of this users guide. The approach taken here is to discuss program flow as it impacts the potential user in an effort to make the flow of information between user and program, and program and user, more easily understood. A Flow Diagram is shown in Figure 1. A sample interaction with the program can be found in Appendix F.

All of the commands needed to begin execution of the program are stored in a CLIST, or Command LIST file. By executing this CLIST file (called GO.LIST), three functions are performed. First, the program is loaded into the computer's memory. Second, the program is given access to all the files it needs for execution. Third, actual execution is begun. The program will not execute if the required files have been archieved. See Appendix G for an approach to de-archiving the files.

The program then proceeds to intiialize itself. Option codes, variable names, parameter codes, and parameter names are read in. The fundamental data set is read in, and the user specific data set is initialized to the NBS supplied sample data set values. Then the error, prompt, help and output files are defined and information is read from each of these files to enable direct access to the appropriate records.

At this point, the first program/user interaction takes place. The program prints out an introduction to the user describing the program, its intended use, and some of the instructions necessary to move successfully between the many modes of program operation. After printing the introduction, the program calls subroutine OPTION. This subroutine prints out the option selections available to the user and then waits for the user to specify one. Depending on the user specification, OPTION will then call one of the working subroutines to perform the required task.

Specification of the change option will cause the calling of the subroutine CHANGE. This routine will ask the user which variable is to be changed and the user will respond with one variable name from either the product specific or the fundamental data set. If the user specifies a product specific variable which is defined by an array of values, the user has the option to change all of the elements in the variable array or just a single element. The user can only change single elements in the variable array for the fundamental data set. Separate subroutines are called, depending on the response, to locate the appropriate data value or array and to prompt for the proper inputs. This procedure is repeated as long as necessary to change all values as required for the next impact calculation. Typing

SOS will return the user to the OPTION subroutine. The user may find it necessary to type SOS twice if the program is in one of the lower level subroutines.

Specification of the input option will cause the calling of the INPUT subroutine. This subroutine will then proceed to prompt the user to input all of the product specific input data, one number at a time until the entire data set is complete. While the program may be directed to return to the OPTION subroutine at any time, it will automatically return at the completion of the input data set.

Specification of the list option will cause the calling of the LIST subroutine. This subroutine will provide a complete list of the total product specific data set, as it exists at the time, on the user's terminal. At the conclusion of the listing the program returns to the OPTION subroutine.

Specification of the print option will cause the calling of the PRINT subroutine. This subroutine will prompt the user to specify which runs are to be printed, and whether the printing should be at the terminal, or if the output should be directed to a file. If the user chooses to output a file, instructions are given concerning how to print that file on the computer center's high speed printer. Output printed on the high speed printer will be mailed to the address specified for the EPAJHT account. When the printing is completely specified, the user is returned to the OPTION level. If the user wishes the output routed to another location, he may do so by using the LIST option and supplementing it with the following commands:

```
tsowyl
COMMAND?
list offline unnumbered remote = nnn box = XXXX
```

where nnn is the identification number of a remote terminal and XXXX is a mailing code. Both of these identifiers can be provided by NCC when an account is established with them.

Selection of the run option will cause the calling of the RUN subroutine. This subroutine sets in motion the steps necessary for the calculation of the impacted population results.

The first step is to call the subroutine ERROR to check for errors in the product specific data file. If any errors exist they are described in error messages on the user's terminal and the program returns to the OPTION subroutine where the errors may be corrected (see p. C-15ff for a listing of the error messages.) If no errors are detected, the subroutine then calls the subroutine PRECAL. This subroutine calculates those parameters necessary and common to all of the various weighting function calculations. The program then returns to RUN where the twelve weighting functions are evaluted, and the impacted populations are accumulated. RUN calls one of three subroutines for the proper weight calculations, ALLEQ for L_{eq} or L_{dn} type calculations, SPEECH for the speech interference calculation, and subroutines are the two sleep disruption weights. Returned from these subroutines are the number of equivalent impacted persons for operators, people exposed in the primary dwelling (non-operators), people exposed in the secondary dwelling, and the total impact of people in the primary dwelling. When all twelve impact calculations have been completed, the subroutine will prompt, asking if the user would like a list of the results on the user's terminal and then return to the OPTION subroutine.

Selection of the stop option will cause the calling of subroutine FINI. This subroutine requests the operator's verification that indeed the session is to be terminated and if the answer is yes, terminates the program execution. If the response is no, the program returns to the OPTION subroutine and asks for the selection of another option.

Selection of the verify option will cause the calling of the VERIFY subroutine. This subroutine will ask the user which variable is to be displayed and the user will respond with one variable name from either the product specific or fundamental data set. If the user specifies a variable which is defined by an array of values the user has the option to look at all the elements in the variable array or just a single element. Separate subroutines are called, depending on the response, to locate the appropriate data values or array. The user may verify as many parameter values as desired. Typing SOS will return the user to the OPTION subroutine. The user may find it necessary to type SOS twice if the program is in one of the lower level subroutines.

If no option is selected, the subroutine OPTION calls ERROR which returns the appropriate error message and then returns to list the options.

As the program moves from level to level in the subroutine structure, offering prompts to the user, the user has some degree of control over the flow through several program direction codes. By typing a question mark (?) the user is presented a list of the assistance codes, which are listed and explained as follows:

HELP,	the user is presented with an expansive explanation of the parameter or input request and then the original prompt is repeated (see p. C-4ff).
CODE,	the user is presented a listing of the parameter subscripts (see p. B-5ff).
SOS,	the user is moved one level up in the program flow. If the program is in the change mode, SOS will return to the option mode, if the program is in a lower level of coding, SOS will return to change, and SOS again will return to option (see note 4 in Appendix G for an example of the use of SOS).

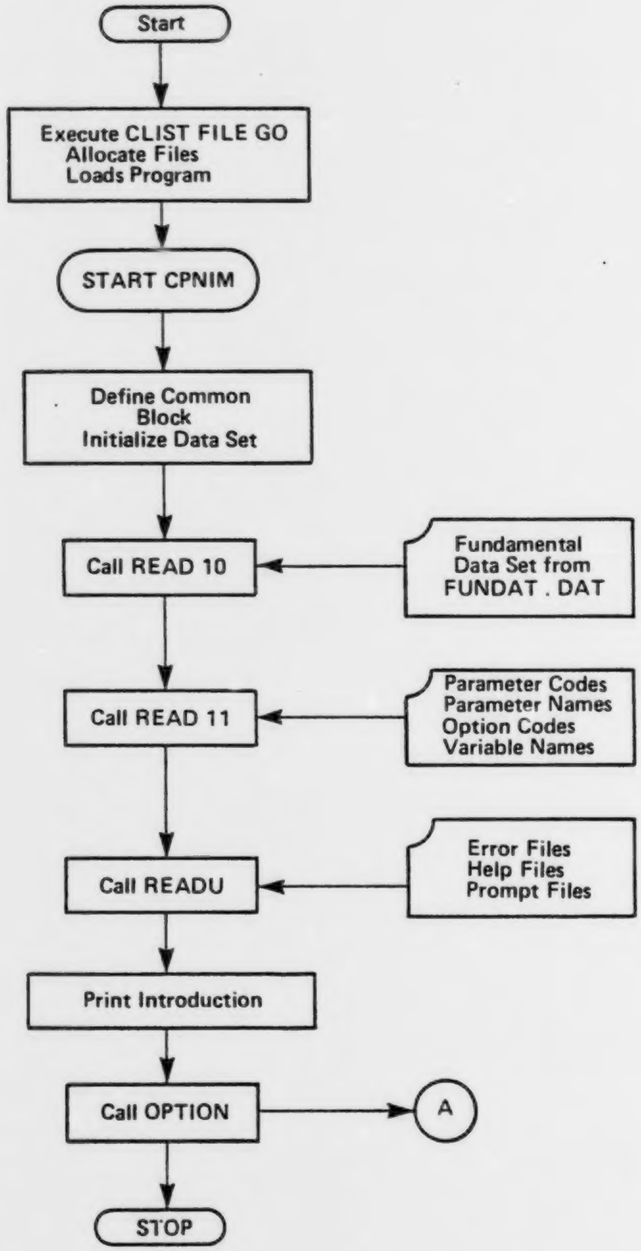
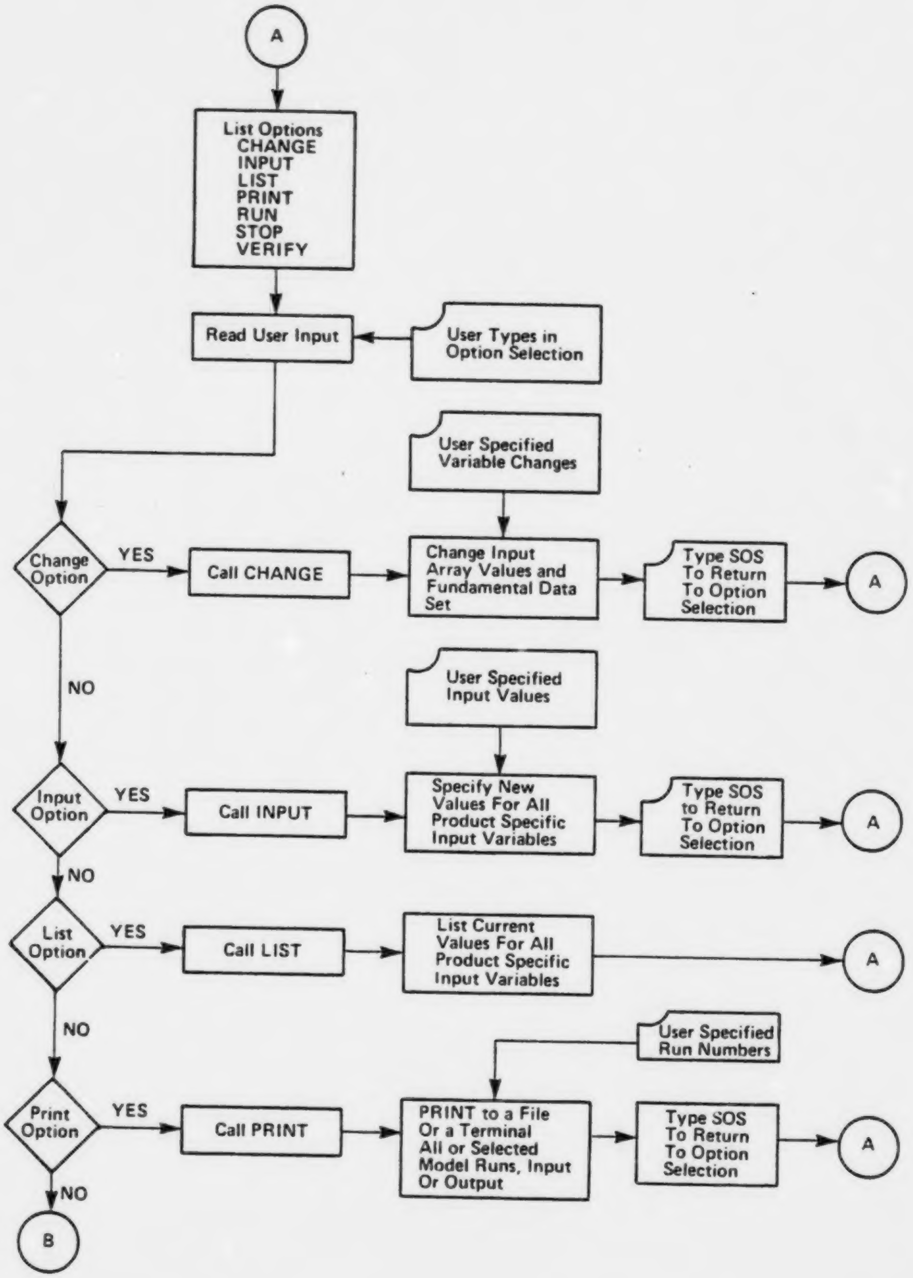
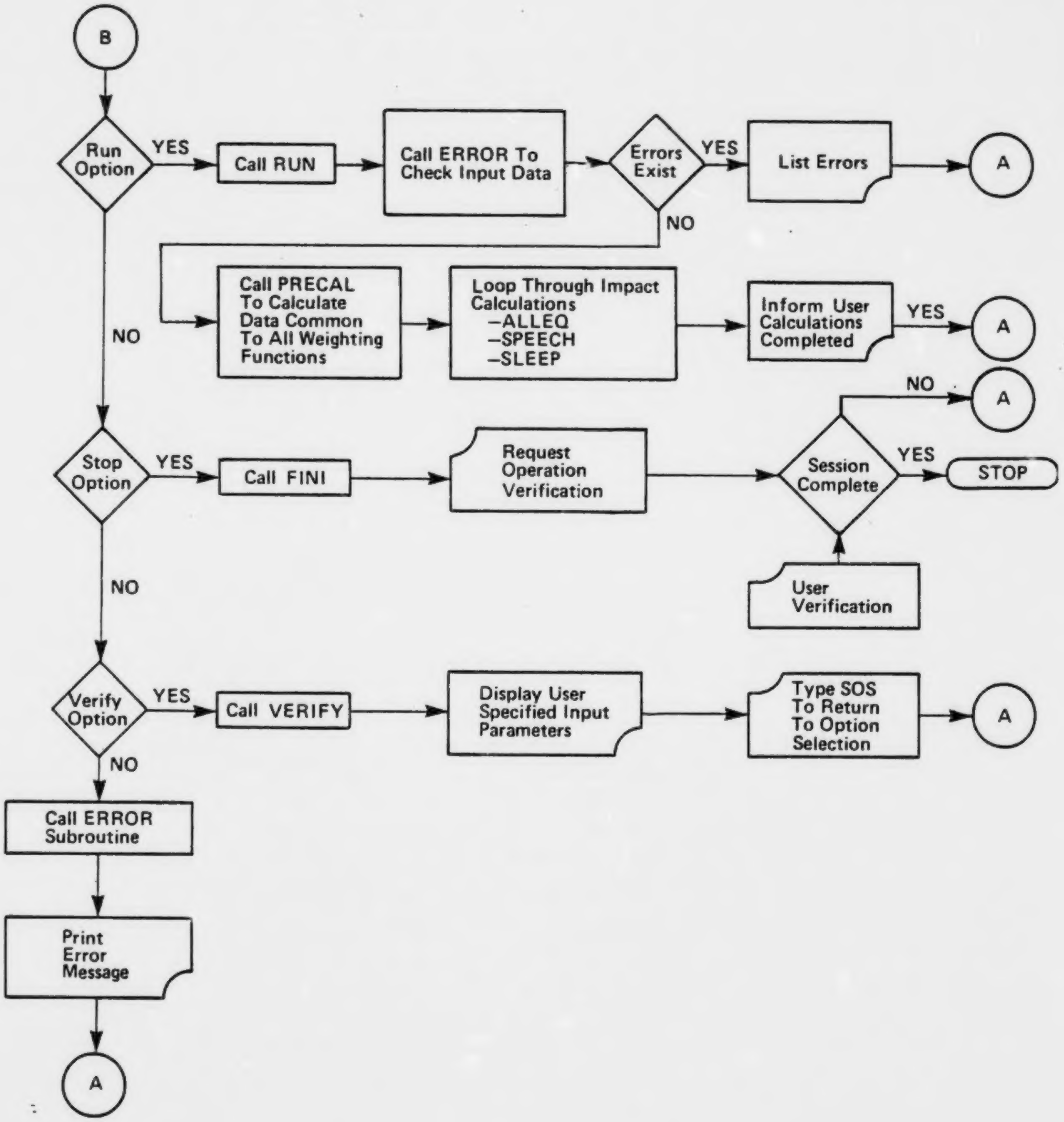

Start
Execute CLIST FILE GO
Allocate Files
Loads Program
START CPNIM
Define Common Block
Initialize Data Set
Call READ 10
Fundamental Data Set from FUNDAT . DAT
Call READ 11
Parameter Codes
Parameter Names
Option Codes
Variable Names
Call READU
Error Files
Help Files
Prompt Files
Print Introduction
Call OPTION
A
STOP

FIG 1. FLOW DIAGRAM OF INTERACTIVE MODE OF CPNIM


A
List Options
CHANGE
INPUT
LIST
PRINT
RUN
STOP
VERIFY
Read User Input
User Types in Option Selection
User Specified Variable Changes
Change Option
YES
Call CHANGE
Change Input Array Values and Fundamental Data Set
Type SOS To Return To Option Selection
A
NO
User Specified Input Values
Input Option
YES
Call INPUT
Specify New Values For All Product Specific Input Variables
Type SOS to Return To Option Selection
A
NO
List Option
YES
Call LIST
List Current Values For All Product Specific Input Variables
A
NO
User Specified Run Numbers
Print Option
YES
Call PRINT
PRINT to a File Or a Terminal All or Selected Model Runs, Input Or Output
Type SOS To Return To Option Selection
A
NO
B


B
Run Option
YES
Call RUN
Call ERROR To Check Input Data
Errors Exist
YES
List Errors
A
NO
Call PRECAL To Calculate Data Common To All Weighting Functions
Loop Through Impact Calculations
–ALLEQ
–SPEECH
–SLEEP
Inform User Calculations Completed
YES
A
NO
A
NO
Stop Option
YES
Call FINI
Request Operation Verification
Session Complete
YES
STOP
User Verification
NO
Verify Option
YES
Call VERIFY
Display User Specified Input Parameters
Type SOS To Return To Option Selection
A
NO
Call ERROR Subroutine
Print Error Message
A

OOPS, allows the user to reenter a single array of data if an error in entry is made. This avoids having to go through change to fix errors made in the input subroutine.

2.1.2 Overview of batch program

The batch version of the program is written to interface directly with many of the subroutines used in the interactive mode of CPNIM. The file BATCH actually contains Job Control Language (JCL) for the operation of the program BACH, which is the controlling program. After the necessary files are defined and allocated to certain unit numbers, BACH begins to execute.

Initial steps define common block and initialize the product specific data to the NBS data set. The program then draws in from already existing files the fundamental data set, the listing of parameter codes, the error messages and prompts, and the product specific input data. These data are read in by calling Read subroutines used in the interactive mode as well, READ10, READ11, and Subroutine READU defines the error, prompt, help and output files, and reads information from each of these files to enable direct access to the appropriate record. Product specific input data is read off of file CPNIM.INPUT through subroutine BACHIN.

The program then calls subroutine ERROR to check for errors in the product specific input data. If any input errors exist, they are all detailed in error messages and placed in the output file. Program execution is terminated and the output file is created. If no errors exist, the program continues and calls subroutine PRECAL which performs basic calculations common to all impact calculations such as calculating the number of exposed persons, the total population, and the normalized noise spectrum.

Impact calculations are then performed by calling the appropriate subroutines. Subroutine ALLEQ calcualtes impact weights for weighting function requiring L_{eq} or L_{dn} types of values. These include Fractional Exposure, Loss of hearing, and General Adverse response calculations. Subroutines SPEECH and SLEEP calculate weighted population figures for the respective impacts. After the calculations are completed, the OUTPUT is sent to a file called CPNIM.OUTPUT. This file is defined through a JCL statement in BATCH. If some other output is desired, BATCH must be changed. Consult an IBM JCL manual for instructions on how to make such changes.

When all weighting functions have been evaluated and the output file is complete the program terminates execution. The output file will contain either error messages, or both the input and output data sets.

Program BATCH flow is shown in the flow diagram in Figure 2.

2.2 Operational Environment

The original coding developed by NBS was designed to operate on a Univac 1108 computer using a Fortran V compiler in a batch mode of operation. The configurations described in this Users Guide have been rewritten to be compatable with the EPA/NCC IBM 370 computer using either the Time Sharing Option (TSO) or WYLBUR operating system. Minor revisions were required to make the programming compatable with a Fortran IV G compiler.

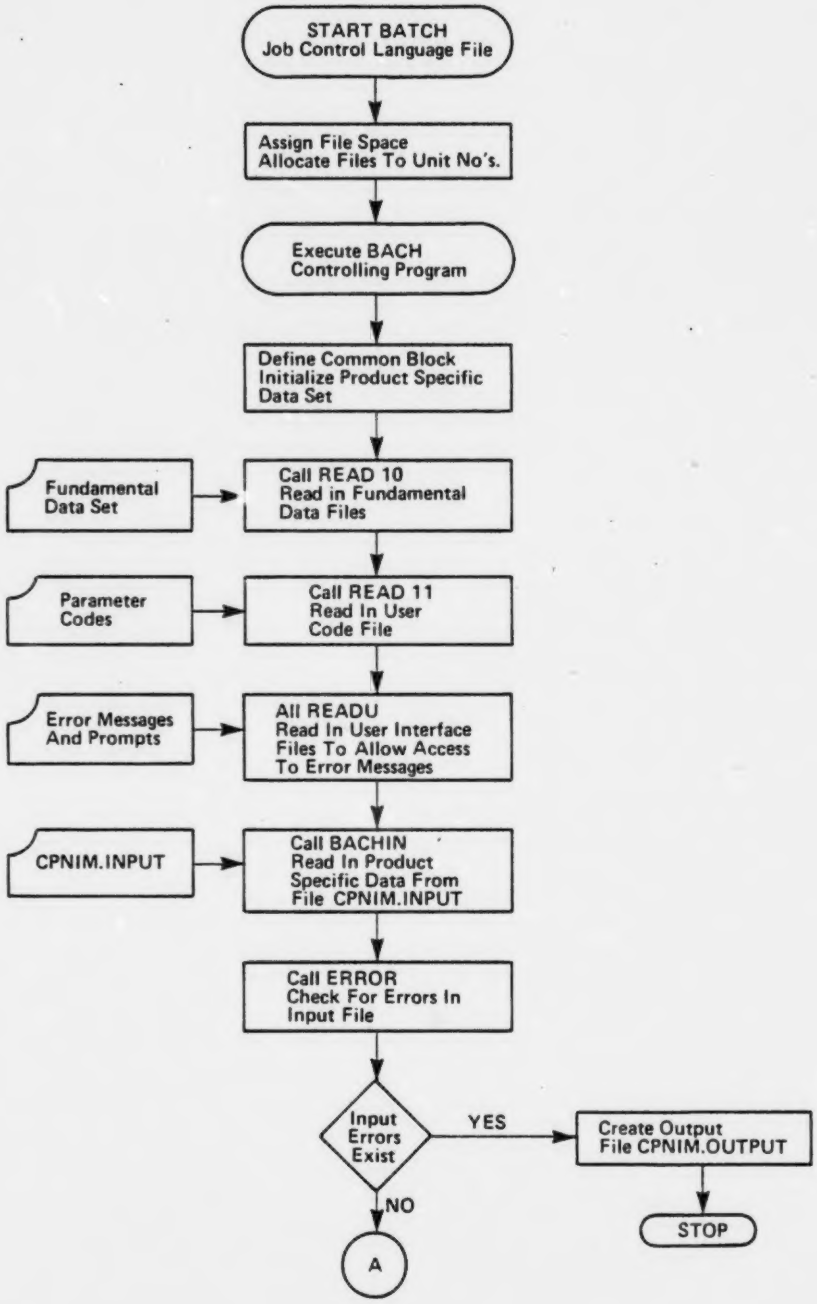

START BATCH
Job Control Language File
Assign File Space
Allocate Files To Unit No's.
Execute BACH
Controlling Program
Define Common Block
Initialize Product Specific
Data Set
Fundamental
Data Set
Call READ 10
Read in Fundamental
Data Files
Parameter
Codes
Call READ 11
Read In User
Code File
Error Messages
And Prompts
All READU
Read In User Interface
Files To Allow Access
To Error Messages
CPNIM.INPUT
Call BACHIN
Read In Product
Specific Data From
File CPNIM.INPUT
Call ERROR
Check For Errors In
Input File
Input
Errors
Exist
YES
Create Output
File CPNIM.OUTPUT
STOP
NO
A

FIG 2. FLOW DIAGRAM OF BATCH MODE OF CPNIM

2.3 System Configuration

This programming is designed to run on EPA's NCC, IBM 370 system. Access to both the interactive and batch programs is through a low speed terminal although output can be obtained on either high speed or low speed terminals. No provisions have been made for card or tape inputs. A default data set is contained in block data within the program structure.

3. SYSTEM OPERATION

3.1 Common System Initiation Procedures

The procedures which must be followed to intiate the CPNIM model in both the interactive and batch modes of operation will be detailed in this section. The interactive mode must be accessed through the TSO system while the batch mode is accessed through the WYLBUR system. Logon procedures for TSO are described in section 3.1.1 and for WYLBUR in section 3.1.2.

If the teriminal to be used is a permanent part of the NCC system, logon will follow normal procedures, and the operator should ignore the initial steps relating to terminal hookup. If a terminal is to be used which is not a permanent part of the system, it must bein the HALF-DUPLEX mode and ON-LINE. The appropriate telephone number for Low Speed System Access is dialed and the phone is placed in the acoustic coupler cradle. For convenience, a list of phone numbers for low speed terminal access has been provided in Table 1.

3.1.1 System logon for the interactive program

The System TSO option is accessed by typing TSO followed by a carriage return. The system will respond with enter LOGON. The user then logs on by typing LOGON EPAJHT/ROBERT. EPAJHT is the user identification code that must be used to connect to the CPNIM program and ROBERT is the currently valid password. The password may be changed by following procedures outlined in the WYLBUR user's guide. When the system responds with READY, the TSO logon is complete.

TABLE 1. TELEPHONE NUMBERS TO CONNECT TO THE NATIONAL COMPUTER CENTER USING A LOW SPEED TERMINAL.

State	City	Telephone
Alabama	Montgomery	(205) 934-3410
California	San Francisco	(415) 986-8200
Colorado	Denver	(303) 572-1107
Connecticut	New Haven	(203) 787-1702
District of Columbia	Washington	(202) 841-9330
Georgia	Atlanta	(404) 659-6670
Illinois	Chicago	(312) 368-4700
Louisiana	New Orleans	(504) 586-1071
Massachusetts	Boston	(617) 482-1854
Michigan	Detroit	(313) 963-3388
Missouri	Kansas City	(816) 677-2833
Nevada	Las Vegas	(702) 293-0300
New York	New York	(212) 532-0437
Ohio	Cincinnati	(513) 791-5311
Pennsylvania	Philadelphia	(215) 567-1381
North Carolina	Raleigh/Durham	(919) 832-6592
South Carolina	Columbia	(803) 252-0840
Tennessee	Nashville	(615) 361-7566
Texas	Dallas	(214) 636-8838
Washington	Seattle	(206) 625-9937

From Users Guide for Interactive Access, NCC, December 1980. For locations not listed here call NCC Communication Control 919-541-4506 or FTS 629-4506.

3.1.2 System logon for the batch program

Entering the WYLBUR System is accomplished by typiing WCCWYL followed by a carriage return. The terminal will respond READY TO WCC ON WYLBUR and then ILLEGAL TERMINAL TYPE. The user must identify the terminal type in use through the appropriate code. The system will then present any system messages and then ask for the logon User Identification (EPAJHT), Account Number (S2KC), and Password (ROBERT). If all three inputs are accepted the system will then request a global format by writing SPECIFY GLOBAL FORMAT FOR SAVE COMMANDS and will provide a list of options. The user should respond with DEFAULT.

The system will respond "LOGON" NOT FOUND IN "WYLIB" ON USER63 and then will type COMMAND. This means that the logon has been accepted and the user is now in the command mode of the WYLBUR System.

3.2 Description of the Interactive Program

The interactive mode of operation is designed to prompt the user for all user supplied inputs and to provide, on request, more detailed descriptions of the inputs. It contains default data shown in Table 2 for all the user supplied data which corresponds to those supplied by the National Bureau of Standards in the original program description, and fundamental data on the movement of persons in dwellings, and room acoustical characteristics. The program allows access to all data for changes and eliminates the need for having the operator become familiar with the program or even the WCC system.

```
******************

 TITLE
 CONSUMER PRODUCT NAME AND OPERATOR REQUIREMENT

 NBS SAMPLE DATA SET

******************

 OPTYP
 FRACTION OF OPERATORS FOR PERSON TYPE

 PERSON TYPE
-----------
 UNEMPLOYED ADULT MALE            0.1000
 EMPLOYED   ADULT MALE            0.0500
 UNEMPLOYED ADULT FEMALE          0.5000
 EMPLOYED   ADULT FEMALE          0.2500
 SCHOOL     AGE   CHILD           0.1000
 PRESCHOOL        CHILD           0.0

******************

 PERC
 FRACTION OF EACH HOUSE THAT OWN THE CONSUMER PRODUCT

 HOUSE TYPE
-----------
 SINGLE FAMILY                    0.9920
 TOWNHOUSE                        0.9920
 MULTIFAMILY                      0.9920

******************

 ONTIM
 MINUTES/DAY THE CONSUMER PRODUCT OPERATES

                     ------------- DAY TYPE -------------
 HOUSE TYPE          WEEKDAY       WEEKDAY       WEEKEND
----------           (SCHOOL)      (NO SCHOOL)
 SINGLE FAMILY         25.70         25.70         25.70
 TOWNHOUSE             19.30         19.30         19.30
 MULTIFAMILY           12.80         12.80         12.80
```

TABLE 2. DEFAULT INPUT VALUES IN CPNIM

```
******************

 TIM
 LENGTH OF TIME (IN MINUTES) THE CONSUMER PRODUCT IS USED
 DURING EACH OPERATION

                       ------------- DAY TYPE -------------
 HOUSE TYPE            WEEKDAY         WEEKDAY        WEEKEND
 ----------            (SCHOOL)        (NO SCHOOL)
 SINGLE FAMILY           60.00           60.00          60.00
 TOWNHOUSE               45.00           45.00          45.00
 MULTIFAMILY             30.00           30.00          30.00

******************

 TP9
 PROBABILITY OF THE CONSUMER PRODUCT BEING OPERATED IN A
 GIVEN TIME PERIOD

------------------

 FOR TIME PERIOD : 700 -  900 HRS

                       ------------- DAY TYPE -------------
 HOUSE TYPE            WEEKDAY         WEEKDAY        WEEKEND
 ----------            (SCHOOL)        (NO SCHOOL)
 SINGLE FAMILY            0.05            0.05           0.05
 TOWNHOUSE                0.05            0.05           0.05
 MULTIFAMILY              0.05            0.05           0.05

------------------

 FOR TIME PERIOD : 900 - 1700 HRS

                       ------------- DAY TYPE -------------
 HOUSE TYPE            WEEKDAY         WEEKDAY        WEEKEND
 ----------            (SCHOOL)        (NO SCHOOL)
 SINGLE FAMILY            0.75            0.75           0.75
 TOWNHOUSE                0.75            0.75           0.75
 MULTIFAMILY              0.75            0.75           0.75

------------------
```

```
 FOR TIME PERIOD :1700 - 2200 HRS

                     ------------- DAY TYPE -------------
 HOUSE TYPE          WEEKDAY        WEEKDAY          WEEKEND
----------           (SCHOOL)       (NO SCHOOL)
 SINGLE FAMILY          0.20           0.20             0.20
 TOWNHOUSE              0.20           0.20             0.20
 MULTIFAMILY            0.20           0.20             0.20

------------------

 FOR TIME PERIOD :2200 -  700 HRS

                     ------------- DAY TYPE -------------
 HOUSE TYPE          WEEKDAY        WEEKDAY          WEEKEND
----------           (SCHOOL)       (NO SCHOOL)
 SINGLE FAMILY          0.0            0.0              0.0
 TOWNHOUSE              0.0            0.0              0.0
 MULTIFAMILY            0.0            0.0              0.0

******************

 RMP9
 PROBABILITY OF THE CONSUMER PRODUCT BEING OPERATED IN A
 GIVEN ROOM

------------------

 FOR SOURCE ROOM :KITCHEN

                     ------------- DAY TYPE -------------
 HOUSE TYPE          WEEKDAY        WEEKDAY          WEEKEND
----------           (SCHOOL)       (NO SCHOOL)
 SINGLE FAMILY          0.08           0.08             0.08
 TOWNHOUSE              0.11           0.11             0.11
 MULTIFAMILY            0.17           0.17             0.17

------------------
```

```
 FOR SOURCE ROOM :LR-DR-FR

                    ------------- DAY TYPE -------------
 HOUSE TYPE         WEEKDAY        WEEKDAY         WEEKEND
----------          (SCHOOL)       (NO SCHOOL)
 SINGLE FAMILY        0.50           0.50            0.50
 TOWNHOUSE            0.56           0.56            0.56
 MULTIFAMILY          0.50           0.50            0.50

-----------------

 FOR SOURCE ROOM :BATHROOM

                    ------------- DAY TYPE -------------
 HOUSE TYPE         WEEKDAY        WEEKDAY         WEEKEND
----------          (SCHOOL)       (NO SCHOOL)
 SINGLE FAMILY        0.0            0.0             0.0
 TOWNHOUSE            0.0            0.0             0.0
 MULTIFAMILY          0.0            0.0             0.0

-----------------

 FOR SOURCE ROOM :BEDROOM

                    ------------- DAY TYPE -------------
 HOUSE TYPE         WEEKDAY        WEEKDAY         WEEKEND
----------          (SCHOOL)       (NO SCHOOL)
 SINGLE FAMILY        0.34           0.34            0.34
 TOWNHOUSE            0.33           0.33            0.33
 MULTIFAMILY          0.33           0.33            0.33

-----------------

 FOR SOURCE ROOM :BASEMENT-UTIL-GAR

                    ------------- DAY TYPE -------------
 HOUSE TYPE         WEEKDAY        WEEKDAY         WEEKEND
----------          (SCHOOL)       (NO SCHOOL)
 SINGLE FAMILY        0.08           0.08            0.08
 TOWNHOUSE            0.0            0.0             0.0
 MULTIFAMILY          0.0            0.0             0.0

-----------------
```

```
 FOR SOURCE ROOM :OUTDOORS

                     ------------- DAY TYPE -------------
 HOUSE TYPE          WEEKDAY        WEEKDAY        WEEKEND
----------           (SCHOOL)       (NO SCHOOL)
 SINGLE FAMILY         0.0            0.0            0.0
 TOWNHOUSE             0.0            0.0            0.0
 MULTIFAMILY           0.0            0.0            0.0

******************

 SPECTR
 OCTAVE BAND LEVELS FOR THE CONSUMER PRODUCT

 OCTAVE BAND  LEVEL(DECIBELS)
-----------
   63 HZ       62.20
  125 HZ       69.50
  250 HZ       71.50
  500 HZ       71.00
 1000 HZ       71.10
 2000 HZ       64.20
 4000 HZ       56.80
 8000 HZ       44.50

******************

 IDB
 STEPSIZE OF THE DECIBEL RANGE

   5 DECIBELS

******************

 MIN
 MINIMUM OF THE DECIBEL RANGE

  70 DECIBELS
```

```
*****************

 MAX
 MAXIMUM OF THE DECIBEL RANGE

  95 DECIBELS

*****************

 OPDIFF
 DIFFERENCE BETWEEN THE SOUND LEVEL AT THE OPERATOR LOCATION
 AND THE POWER LEVEL IN THE SOURCE ROOM

 -12.000 DECIBELS

*****************

 DISLEV
 THE PROBABILITY OF THE POWER LEVEL IN THE SOURCE ROOM BEING
 IN A PARTICULAR DECIBEL RANGE

 DB RANGE
-----------
  70 -  75 DB      0.190
  75 -  80 DB      0.330
  80 -  85 DB      0.190
  85 -  90 DB      0.150
  90 -  95 DB      0.140

*****************




***************
```

Unlike the BATCH mode of operation, in this mode any data sets built or changed are only temporary. When the program is terminated, all files are cleared and any restart begins with the default values.

This subsection will describe the manner in which a user interfaces with the CPNIM interactive model.

3.2.1 Security

No security beyond normal identification procedures has been supplied wtih this program.

3.2.2 Initiation procedures

After successful logon to the NCC TSO system, the program CPNIM is initiated by typing:

```
EXEC GO.
```

GO contains all of the necessary Job Control Language to run the program and also allocates units for the storage and retrieval of data blocks. The system will, after a short period of time, respond with an introduction and some instructions on the means which an operator has for moving between the various modes of program operation. From this point on, the operator interacts with the program to input data, check or list inputs, change data, verify the data for particular inputs, run the impact calculations, view the results or print the results at a high speed printer. This interaction is explained in the following sections.

3.2.3 Operational procedures

While the program may be run without the generation of a user supplied input data file, the normal operating procedure would be to start the program, create an input data set through the input option, view the data through the list option, change any errors using the change option, and then run the impact calculations through the run option. At this point the output data can be viewed, or listed for later analysis and the program is stopped or another run is begun using the same procedure.

The operator interfaces with the program by responding to the prompts with data or instructions, and by using the program guide commands:

SOS - Returns to next subroutine level

OOPS - Returns to last prompt

CODE - Shows parameter subscripts

HELP - Shows appropriate Help message

? - Shows program guide commands

The program is written in three basic phases, data input and manipulation, impact calculation, and display of results. In moving between these phases an attempt has been made to foresee as many of the potential pitfalls as possible and provide program traps and appropriate error messages. Where any instructions could appear unclear, help messages have been provided although there are not help messages for each step of the program. See Appendix G for examples of error messages. See pages C-4 through C-14 for a complete list of the available help messages; pages C-14 through C-17 provide a complete list of the error messages.

3.2.4 Error and recovery procedures

Because the program is written to detect errors in input data, there is little chance that a program usage error could result in the programs failing to execute. In general errors will be returned to the operator and can be corrected without leaving the program flow through the CHANGE subroutine. If a catastrophic error should occur, and the program terminates itself, the only recourse is to restart the program, check the input variables very carefully, and then rerun the program.

In as much as the program is successfully operating on the NCC system at the time of this writing, any error which is not immediately traceable directly to an error in input data must reflect some system change at NCC. If such is the case, operation of the CPNIM program will not be possible until the change has been identified and the program has been adjusted accordingly. Reference to the NCC user's manual or a telephone call to user support should be the first step to remedy such a problem.

3.2.5 Constraints and limitations

The interactive mode of the CPNIM program is designed to operate on only one set of input data at a time. Once the data is input and the impact calculations have been completed, other runs may be made before the output files are listed but only one input file may be generated at a time. If the user is inadvertently logged off of the system during a run, all input and output files will not be lost. The output will be appended to any existing output files in CPNIM.OUTPUT which can be listed or deleted at any time.

Appendix B.1 specifies restrictions on product specific variables, and the definitions of input parameter codes in Appendix B.3 make the limitations on them evident. An examination of the prompt, help, error files in Appendix C will give the user a clear idea of the help he may expect from these messages. An examination of the comment cards (a "C" appears in the first column of the line) may provide useful information. For example, lines 27-28 indicate that a temporary file is created to take care of outputs and inputs for as many as twenty runs. This means that the user may not expect to perform more than twenty runs for any one logon to logoff period. This is hardly an onerous limitation, but it is a limitation nonetheless. Other limitations on the format and content of input and output can be found in the sections that deal with those matters.

Regardless of what happens during the running of the CPNIM program, all input files are deleted when the program session is ended. This allows the user to start each session knowing that the input files are clear. CPNIM.OUTPUT must be deleted if used.

3.2.6 Data base

The CPNIM program requires a very large data base to properly calculate the population impacted figures which are output. To reduce the user workload in operating the program the data requirements have been broken down into two categories. The user inputs product specific data, that is information pertaining only to the product noise, ownership, and usage patterns. This data should be identified by the user before the program is set in motion. The program prompts the user for each data input. These data are shown in Table 2 along with the default values already in the program in block data.

The program also uses a large block of data on the movement of persons within a dwelling and on the noise reduction characteristics of rooms in dwellings. These data are contained in the data file FUNDAT.DAT and are shown in Appendix E. This file has been copied directly from the data supplied by NBS in the original program and while it may not be totally accurate, it is a good starting point for product ranking assessment. As better data on these parameters becomes available it can be incorporated into the model.

3.2.7 Control language

The interactive version of the CPNIM program has been written so that the only control language necessary to start the program is to type EXEC GO. The CLIST GO contains all of the necessary commands to direct program execution and is shown here only to provide some insight into what Job Control Commands are actually necessary to operate the model.

```
00010 ATTRIB FOOB RECFM(F B) BLKSIZE(4000) LRECL(80)
00020 ALLOC FI(FT05F001) DA(*)
00030 ALLOC FI(FT06F001) DA(*)
00040 ALLOC FI(FT09F001) NEW
00050 ALLOC FI(FT10F001) DA(FUND.DAT)
00060 ALLOC FI(FT11F001) DA(CODDAT.DAT)
00070 ALLOC FI(FT30F001) NEW
00080 ALLOC FI(FT50F001) DA(CPNIM.PROMPT)
00090 ALLOC FI(FT51F001) DA(CPNIM.HELP)
00100 ALLOC FI(FT52F001) DA(CPNIM.ERROR)
00110 ALLOC FI(FT66F001) DA(CPNIM.OUTPUT) MOD USING(FOOB)
00120 LOAD (LOADLIB(MAIN)) EP(MAIN)
00130 FREE ALL
END OF DATA
```

3.2.8 Input procedures

Input procedures and program interaction are very easy. Once the data to be input has been defined (see Appendix B) and the program has entered the option level, the input option is selected. The program will begin by prompting for a title and operator requirement. While any title may be used as long as it is 72 characters or less, normally this line will contain the product name and an indication of whether an operator is required.

The program will then work its way through all of the product specific data one point at a time. For data where there is a possibility for dependence on other parameters, [i.e. the length of time the consumer product is used (TIM) during each operation may be considered to be a function of the type of day] the program will ask for such a dependence. If there is no dependence, the program will only ask for the minimum data input and then use that data to fill the rest of the array. This situation is displayed in Table 2. In many of the default input data sets numbers are repeated. If no data is input for a particular parameter, then the default data is used.

At any time during the input of data through the input subroutine, a more expansive description of the prompt may be viewed by typing HELP. If a help message is available it will be printed and the prompt will be repeated. When the entire data set is entered the program will return to the option point and print out the available options. The inputing of data may be terminated at any time by typing SOS, although it should be remembered that during the course of a use session, the input data set remains as it was last set. If no inputs are provided the data is the default value. If one run is made where the values are changed, then the default values for the remainder of the session will be those input values until they are changed again. The only way to return to the original default values would be to end the session and start over again.

If a large part of the default data set (or of the existing data set) is to be used for the next run then only a few variables can be changed by using the change option. The program will ask which variable is to be changed and the user must respond with the correct variable name. Typing help at this point will result in a listing of the variable names but the user must be knowledgeable enough to select the correct one. The user may change any of the user supplied, product specific variables desired through this option. It is also possible to specify the changing of the fundamental data but only one value at a time can be changed which tends to lead to a rather long and involved process considering the large amount of data in the set. As with input, none of the changes made will appear after the session is terminated.

If it is unclear to the user what data is currently in the input data file, several options are available. By using the list option, all of the product specific input data may be listed at the operator's terminal. This will, however, take several minutes and a lot of terminal paper so it may be more expedient to use the verify option. The verify option will ask which variable to display and then, if the variable is an array, will ask whether the user wishes to view any specific value, or all of them. By only verifying a few variables much time can be saved.

3.2.9 Output of the interactive mode

After specifying the RUN option the program will calculate all of the weighting functions and return to inform the user that the computations are completed. The program will then ask whether the user wishes to view the output or to store it for later print out. If the output is viewed at this time, it will still exist as a numbered and titled output file for output at another time during the use session. Be sure to remember to view the output files before terminating the session as they will be deleted when the session is ended.

If the user answers no to the prompt, the program returns to the option level where the user may elect to enter new data for another run or select print to show the results of any of the previous runs in the session. In response to selecting the print option the program will ask which of the existing output files is to be printed and lists all that have been created. The user picks one of the files by number. The program then asks whether the output should appear at the user's terminal or be placed on a file to be printed later.

If the printout is to be to a file, the program will give a lengthy prompt as to which file has been created and will explain how to get the file listed either at the user's terminal or off line as explained in Section 3.3.3. The file created will be CPNIM.OUTPUT which should be deleted at sessions end by typing DELETE CPNIM.OUTPUT in TSO command mode or by using the SCRATCH command in WYLBUR.

If the printout is to be at the user's terminal then the program will ask if the listing should show the product specific inputs as well. After the user answers this question, the output will be listed. The output, if it contains the input listing, will show the product specific input variables as in Table 2. The weighted population impacted figures for the Twelve weighting functions will follow the title and a value which represents the number of people exposed to the product.

The first line of output data will be the title which should contain the product name and the operator requirement. The next two lines will show the total number of persons exposed to the product's noise and the total population of the United States.

The first weighting function shown is a fractional exposure type of calculation based on an L_{eq} of 70 dB. This weighting was chosen because the majority of consumer products do not, in and of themselves, present significant noise hazards. They do, however, contribute to the overall daily noise dose and this weighting is an indication of the extent of that contribution. The output Weighted Population is actually the population exposed to the product noise multiplied by the Weighting function. In this case the weight is upper bounded to be one at an L_{eq} of 70 dB so that exposures of greater than 70 dB do not produce larger weighted populations. The Noise Impact Index is the Weighted Population divided by the population at risk.

The next two weighting functions are designed to point out impacts of products which do, in and of themselves produce significant noise dose problems in some type of Noise Induced Permanent Threshold Shift. The first weighting is based on the expected average NIPTS at the four major audiometric frequencies over a 40-year period of exposure and is set to zero for exposures of L_{eq} less than 70 dB. The second weight estimates the NIPTS at 4,000 HZ, an area of the spectrum most susceptable to loss of hearing acuity. For both weighting functions, the weighted population represents the number of people exposed to the product noise multiplied by the average number of NIPTS.

Weighting function four is based on the relationship between general adverse response and the yearly, Day-Night, average, sound level. This weighting function has been normalized to one at an L_{dn} of 75 dB and, as such, does not produce a weighted population which represents the predicted number of highly annoyed persons, although studies of the number of highly annoyed people exposed to outdoor noise levels formed the basis for the weighting function. Weight five is the same function with a + 15 dB penalty applied for noises produced indoors. Either weight, when used to rank order products, will produce the same ranking.

The sixth weighting function expresses the general adverse response to day-night noise exposures from outdoor noise sources in terms of the Fractional Impact, a method used in previous EPA analysis. The function is a straight line fit to the curve represented by the fourth weight. This weight is zero for values of L_{dn} less than 55.

Weighting functions seven, eight, and nine have been included to represent work presently being done on noise exposure in other industrialized countries. These three weights are based on work by Alexandre and Barde where fractional impact is assessed through functions which parallel the growth in acoustical energy, as in the Fractional Exposure weight, and loudness and a function which attempts to average between the two.

The tenth weighting function is based on speech interference and is calculated from the fraction of speech unintellegibility at a given sound level and the amount of exposure time. The intensity of speech interference is a measure of the fraction of sentences lost during an average day.

The last two weighting functions represent potential disturbances to sleep from Sound Exposure Level and are only calculated for exposures between the hours of 10 PM and 7 AM. The first weight reflects the potential for sleep disruptions, defined as a change from one sleep state to another, while the second represents total awakening. The integration time for SEL is limited to two minutes, as it is felt that if awakening or arousal has not occurred within that time period, it is unlikely to occur at all.

More in depth explanations for each of these weighting functions can be found in the report on the original model programmed by NBS [*1*].

3.3 Description of Batch Program

This mode of operation is designed to provide a less expensive alternative to the interactive program for operators who are familiar with the WCC WYLBUR system. It can operate at one quarter the cost of the interactive model but turn around time will be a stronger function of the load on the system. This section details the use of the "Batch" program and gives some guidance in the input of the required, product specific variables. Fundamental data may be changed through normal WYLBUR commands as well, but care should be exercised in doing so, so as not to inadvertently, permanently alter the fundamental data.

It is well beyond the scope of this user's guide to provide an adequate tutorial on the WYLBUR system and as such, the operator is referred to the system manual for guidance with error messages and a more detailed description of data management techniques.

The "Batch" program makes use of several of the subroutines written for CPNIM. The program itself, shown in Section 2.1.2, is simply organizational, calling the proper subroutines at the appropriate time. The only file not available to the program is the input file called "CPNIM.INPUT." This file provides the interface between the program and the user and is initialized to the NBS supplied, product specific data for vacuum cleaners. [*1*] The user will build or change this file to provide the product specific data necessary for a given run through standard WYLBUR commands.

When the "INPUT" file is completed and saved, the job may be submitted and run as a typical WYLBUR job.

3.3.1 Security

No security beyond normal identification procedures has been supplied with this program.

3.3.2 Initiation procedures

The batch program will run without any user supplied product specific data as default values have been provided which reflect the original NBS data set for vacuum cleaners. This data is shown in Table 2. It is however, important to create the input data file even if it is empty. The procedures in this section detail the input of product specific data, the running of the model, and the printing of the output data file.

3.3.2.1 Creation of input file

After successful logon to the WCC Wylbur system the operator must create the input data file CPNIM.INPUT after scratching any old input files. To create any file the system must be in the COLLECT mode as opposed to the COMMAND mode. One may enter the COLLECT mode from COMMAND by typing any of the following:

- Hitting the BREAK key,
- COLLECT,
- or COLLECT N where N is the line you wish to enter.

Once in COLLECT the file is generated in a free format. That is, the data is entered serially, separated by a comma. The first line of data is a 72 character title, starting in column one, which can say anything but should normally contain the product name and whether an operator is required or not. The second line, and all those following it, must start in column two. The second line defines the set as NAMELIST INS and appears as:

```
1 2 3 4 5
_ & I N S
```

The variables to be input are described below.

OPTYP Six numbers representing the probability that a given person type is operating the product. The person types are: Unemployed Adult Male, Employed Adult Male, Unemployed Adult Female, Employed Adult Female, School Age Child, Pre School Child. The default values are:

```
__OPTYP=.1,.05,.5,.25,.1,0.0
```

PERC Three numbers representing the percentage of households of a certain type owning the product. The dwelling types are: Single Family, Townhouse, Apartment. The default values are:

```
___PERC=.992,.992,.992
```

ONTIM Nine numbers representing the number of minutes per day of product usage, based on a yearly average for the three dwelling types on the three different day types. Dwelling types are defined as above, day types are Weekday with School, Weekday without School, Weekend. The default values are:

```
__ONTIM=25.7,19.3,12.8,25.7,19.3,12.8,25.7,19.3,12.8

          Day Type 1     Day Type 2     Day Type 3
```

Note: The above values reflect the conclusions the operation of a vacuum cleaner is independent of day type. Whereas the interactive mode of program operation will fill in blanks for such a condition, requiring the entering of only three numbers, all numbers must be entered into the file for batch operation or the default values will be used for any numbers not entered. Where the default values differ from the intended values, errors will result.

TIM Nine numbers representing the total amount of time, in minutes, that a product is operated whenever it is turned on in a given dwelling type on a particular day type as defined above. The default values are:

__TIM=60.,45.,30.,60.,45.,30.,60.,45.,30.

TP9 Thirty six numbers representing the probability of the product being operated in the four time periods: 7 to 9 AM, 9 AM to 5 PM, 5 to 10 PM, and 10 PM to 7 AM, in the three dwelling types, on the three day types. The default values are:

__TP9=.05,.75,.2,0.0,.05,.75,.2,0.0,.05,.75,.2,0.0

House Type 1 2 3

Day Type 1

___.05,.75,.2,0.0,.05,.75,.2,0.0,.05,.75,.2,0.0

House Type 1 2 3

Day Type 2

___.05,.75,.2,0.0,.05,.75,.2,0.0,.05,.75,.2,0.0

House Type 1

Day Type 3

Note: House types 1, 2, and 3, and Day types 1, 2, and 3 are the same, reflecting the conclusions that the probability of operation of the product in a time period is independent of house type and day type. All 36 data points must be entered or the default values will be used in computations leading to errored results.

RMP9 Fifty-four numbers representing the probability of the product being operated in the six room types: Kitchen, Living/Dining/Family room, Bathroom, Bedroom, Basement/Shop, Outdoors, in the three dwelling types, for the three day types. The default values are:

```
__RMP9=.08,.5,0.0,.34,.08,.00,.11,.56,.00,.33,.00,.00
       Dwelling Type 1                    2
       .17,.50,.00,.33,.00,.00
                 3
                            Day Type 1

    ___.08,.5,0.0,.34,.08,.00,.11,.56,.00,.33,.00,.00
       Dwelling Type 1                    2
       .17,.50,.00,.33,.00,.00
                 3
                        Day Type 2

    ___.08,.5,0.0,.34,.08,.00,.11,.56,.00,.33,.00,.00
       Dwelling Type 1                    2
       .17,.50,.00,.33,.00,.00
                 3
                        Day Type 3
```

	Note: Day types 1, 2, and 3 are the same, reflecting the conclusions that the probability of operation of the product in a given room is independent of the day type. All 54 data points must be entered or the default values will be used in computations leading to errored results.
SPECTR	Eight numbers representing linear sound power levels in octave bands from 63 HZ to 8000 HZ. The default values are: __SPECTR=62.2,69.5,71.5,71.0,71.1,64.2,56.8,44.5
IDB	One number representing the number of decibels in each step of the distribution of A-Weighted Sound levels as measured for the product. The default value is: __IDB=5
MIN	One number representing the low value in the first distribution step. The default value is: __MIN=70
MAX	One number representing the high value in the last distribution step. The default value is: __MAX=95
OPDIFF	Three numbers representing the difference between the measured Sound Power Level and the Operator Sound Level. The first number should be calculated based on operator distance, and room acoustical

characteristics. The second two are always zero. The default values are:

```
__OPDIFF=-12.0,0.0,0.0
```

Note: OPDIFF may be calculated from the equation

$$\mathrm{OPDIFF}=10\ \mathrm{Log}_{10}\left(\frac{1}{4\pi R^2}+\frac{4}{A}\right)+10\ \mathrm{dB}$$

Where: R is the distance, in feet, between the operator and the product,
A is the acoustic constant of the room in which the product is most often used.
A = 35 for a Bathroom
A = 45 for a Kitchen or Workroom and
A = 175 for any other room

A good average number is -5 dB for products used indoors.

DISLEV (MAX-MIN)/IDB numbers representing the probability of the Sound Level from the product being in a given level band. The default value is:

```
__DISLEV=.19,.33,.19,.15,.14
```

Note: Provisions have been made for twenty level bands. If more than this is needed, the array will have to be redimensioned.

The last line in the data file is:

```
1 2 3 4 5
_ & E N D
```

If the default values are to be used, it is important to create CPNIM.INPUT anyway. The first two lines and the last line are all that are necessary. If the file contains the data without any errors, it may be saved into the file by typing BREAK to return to the COMMAND mode and then typing:

```
SAVE__CPNIM.INPUT__CARD
```

If a volume is requested type:

```
USER63
```

Now all the necessary files exist to run the model in the batch mode of operation. If a previous run has been made, the output data file will exist as it is generated by the program. To delete the file type:

```
SCRATCH__CPNIM.OUTPUT
```

3.3.3 Bath mode operational procedure

Once the input file has been created and any previous output files have been deleted, the program may be run. In the COMMAND mode of operation type:

```
USE WYLIB (BATCH)
RUN
```

The run command starts the model. The WYLBUR system will assign a job number to the run and place it in the execution que. By typing:

```
LOC__XXXX
```

where XXX is the job number, the operator may find out whether the job is

1) Awaiting Execution,
2) Executing,
3) In Output Hold.

If the latter is the case, an output file CPNIM.OUTPUT has been created. To print the results, two options have been provided. By typing:

```
USE__CPNIM.OUTPUT
L__UNN
```

the operator may view the data at the terminal without destroying the file. If a hard copy is then required, a high speed printer copy may be generated by typing:

```
USE__CPNIM.OUTPUT
L__OFFLINE__BOX MJHT
```

The high speed printer output will be mailed to the EPA project office designated as receiving mail for the JHT account. Once the file has been printed, it is best deleted to avoid future conflicts. If the output data is to be saved, the file can be renamed using standard WYLBUR commands. Old runs may be purged from the system by typing:

```
PURGE__XXXX
```

This allows the command LOC to show only the job of interest. Old jobs are purged daily without any command.

3.3.4 Error and recovery procedures

Only two errors in the operation of this program will cause it to fail to operate. If the input file has not been created or if a previous output file still exists, the program will fail to execute. System response will be to request a VOLUME when CPNIM.OUTPUT is listed. This request indicates that no CPNIM.OUTPUT file exists to recover, correct the problem and re-run. If the old CPNIM.OUTPUT file has not been scratched, it will be the only one available for listing as the other output file will not be accessible.

There are several data checks written into the program to ensure that the proper data are input. These checks are performed in the organizational part of the program prior to any calculations and any errors will generate messages which will appear in the output file CPNIM.OUTPUT. The program will not run completely.

To determine which input values have been put in incorrectly, list out CPNIM.OUTPUT. When CPNIM.INPUT has been corrected, the program can be run again. These checks are detailed in the discussion of the subroutine ERROR.

3.3.6 Constraints and limitations

BATCH may only be operated for one product at a time, and only one run may be executing at any given time.

3.3.7 Control language

Job Control Language for the execution of the program BACH is contained in the file BATCH. The only control necessary to run the program is to run BATCH.

3.3.8 Inputs

Product specific input data are input through the creation of the file CPNIM.INPUT. A default data set is contained in block data. These data are described in Section 3.3.2. Data on the movement of people in dwellings and the acoustical room characteristics of dwellings are contained in the fundamental data set described in Section 3.2.8.

The input file is not automatically deleted after a run. If all of the input data for a following run is to be different from that run just completed, it is best to delete the file CPNIM.INPUT and start over again. If only small changes in product specific variables are to be made, the file may be altered using standard WYLBUR commands and then saved by replacement into the old file by typing:

```
SAVE__CPNIM.INPUT__REP
```

Input data for vacuum cleaners is shown in Table 2. This data has been chosen as the default value and represents the original input data set written by NBS.

3.3.9 Outputs

The output shown when CPNIM.OUTPUT is listed will look exactly like that from the interactive program but the input variables used will be listed as well. If no input errors are found and the program executes properly the output will appear as shown in Table 3.

The first line of output data will be the title which should contain the product name and the operator requirement. The next two lines will show the total number of persons exposed to the products noise and the total population of the United States.

The first weighting function shown is a fractional exposure type of calculation based on an L_{eq} of 70 dB. This weighting was chosen because the majority of consumer products do not, in and of themselves, present significant noise hazards. They do, however, contribute to the overall daily noise dose and this weighting is an indication of the extent of that contribution. The output Weighted Population is actually the population exposed to the product noise multiplied by the Weighting function. In this case the weight is upper bounded to be one at an L_{eq} of 70 dB so that exposures of greater than 70 dB do not produce larger weighted populations. The Noise Impact Index is the Weighted Population divided by the population at risk.

The next two weighting functions are designated to point out impacts of products which do, in and of themselves produce significant noise dose problems in some type of Noise Induced Permanent Threshold Shift. The first weighting is based on the expected average NIPTS at the four major audiometric frequencies over a 40-year period of exposure and is set to zero for exposures of L_{eq} less than 70 dB. The second weight estimates the

```
NBS SAMPLE DATA SET

EXPOSED PERSONS = .2080E+09
TOTAL POPULATION= .2097E+09


----------------

1 FRACTIONAL EXPOSURE WEIGHTING BASED ON AN LEQ OF 70

                          OPERATORS   PRIMARY     SECONDARY   OPERATORS
                                      DWELLING    DWELLING    + PRIMARY
                          ----------  ----------  ----------  ----------
WEIGHTED POPULATION        .7343E+06   .3606E+07   .1967E+04   .4340E+07
NOISE IMPACT INDEX         .3530E-02   .1733E-01   .9458E-05   .2086E-01

----------------

2 POPULATION WEIGHTED LOSS OF HEARING BASED ON CHABA CRITERION

                          OPERATORS   PRIMARY     SECONDARY   OPERATORS
                                      DWELLING    DWELLING    + PRIMARY
                          ----------  ----------  ----------  ----------
WEIGHTED POPULATION        .0          .0          .0          .0
AVERAGE  NIPTS             .0          .0          .0          .0

----------------

3 POPULATION WEIGHTED LOSS OF HEARING AT 4000HZ

                          OPERATORS   PRIMARY     SECONDARY   OPERATORS
                                      DWELLING    DWELLING    + PRIMARY
                          ----------  ----------  ----------  ----------
WEIGHTED POPULATION        .0          .0          .0          .0
AVERAGE NIPTS AT 4000HZ    .0          .0          .0          .0

----------------

4 GENERAL ADVERSE RESPONSE CHABA OUTDOOR NOISE SOURCE

                          OPERATORS   PRIMARY     SECONDARY   OPERATORS
                                      DWELLING    DWELLING    + PRIMARY
                          ----------  ----------  ----------  ----------
WEIGHTED POPULATION        .4395E+07   .1335E+08   .1019E+06   .1478E+08
NOISE IMPACT INDEX         .2113E-01   .6415E-01   .4900E-03   .7103E-01

----------------
```

TABLE 3. INPUT OUTPUT VALUES FROM CPNIM DEFAULT

NIPTS at 4,000 Hz, an area of the spectrum most susceptable to loss of hearing acuity. For both weighting functions, the weighted population represents the number of people exposed to the product noise multiplied by the average number of NIPTS.

Weighting function four is based on the relationship between general adverse response and the yearly, Day-Night, average, sound level. This weighting function has been normalized to one at an L_{dn} of 75 dB and, as such, does not produce a weighted population which represents the predicted number of highly annoyed persons, although studies of the number of highly annoyed people exposed to outdoor noise levels formed the basis for the weighting function. Weight five is the same function with a +15 dB penalty applied for noises produced indoors. Either weight, when used to rank order products, will produce the same ranking.

The sixth weighting function expresses the general adverse response to day-night noise exposures from outdoor noise scurces in terms of the Fractional Impact, a method used in previous EPA analysis. The function is a straight line fit to the curve represented by the fourth weight. This weight is zero for values of L_{dn} less than 55.

Weighting functions seven, eight, and nine have been inclused to represent work presently being done on noise exposure in other industrialized countries. These three weights are based on work by Alexandre and Barde where fractional impact is assessed through functions which parallel the growth in acoustical energy, as in the Fractional Exposure weight, and loudness and a function which attempts to average between the two.

The tenth weighting function is based on speech interference and is calculated from the fraction of speech unintellegibility at a given sound level and the amount of exposure time. The intensity of speech interference is a measure of the fraction of sentences lost during an average day.

The last two weighting functions represent potential disturbances to sleep from Sound Exposure Level and are only calculated for exposures between the hours of 10 PM and 7 AM. The first weight reflects the potential for sleep disruptions, defined as a change from one sleep state to another, while the second represents total awakening. The integration time for SEL is limited to two minutes, as it is felt that if awakening or arousal has not occurred within that time period, it is unlikely to occur at all.

More in depth explanations for each of these weighting functions can be found in the report on the original model programmed by NBS [*1*].

```
5 GENERAL ADVERSE RESPONSE CHABA INDOOR NOISE SOURCE

                          OPERATORS   PRIMARY     SECONDARY   OPERATORS
                                      DWELLING    DWELLING    + PRIMARY
                          ----------  ----------  ----------  ----------
WEIGHTED POPULATION        .3176E+08   .7731E+08   .1231E+07   .8304E+08
NOISE IMPACT INDEX         .1527E+00   .3716E+00   .5917E-02   .3992E+00

----------------

6 GENERAL ADVERSE RESPONSE , FRACTIONAL IMPACT METHOD

                          OPERATORS   PRIMARY     SECONDARY   OPERATORS
                                      DWELLING    DWELLING    + PRIMARY
                          ----------  ----------  ----------  ----------
WEIGHTED POPULATION        .4284E+06   .5764E+07   .0          .7231E+07
NOISE IMPACT INDEX         .2059E-02   .2771E-01   .0          .3476E-01

----------------

7 GENERAL ADVERSE RESPONSE , ALEXANDER ENERGY

                          OPERATORS   PRIMARY     SECONDARY   OPERATORS
                                      DWELLING    DWELLING    + PRIMARY
                          ----------  ----------  ----------  ----------
WEIGHTED POPULATION        .2619E+05   .4607E+06   .0          .6380E+06
NOISE IMPACT INDEX         .1259E-03   .2215E-02   .0          .3067E-02

----------------

8 GENERAL ADVERSE RESPONSE , ALEXANDER LOUDNESS

                          OPERATORS   PRIMARY     SECONDARY   OPERATORS
                                      DWELLING    DWELLING    + PRIMARY
                          ----------  ----------  ----------  ----------
WEIGHTED POPULATION        .2150E+06   .3097E+07   .0          .3980E+07
NOISE IMPACT INDEX         .1033E-02   .1489E-01   .0          .1913E-01

----------------

9 GENERAL ADVERSE RESPONSE , ALEXANDER SYNTHESIS

                          OPERATORS   PRIMARY     SECONDARY   OPERATORS
                                      DWELLING    DWELLING    + PRIMARY
                          ----------  ----------  ----------  ----------
WEIGHTED POPULATION        .1260E+06   .1938E+07   .0          .2548E+07
NOISE IMPACT INDEX         .6059E-03   .9318E-02   .0          .1225E-01

----------------
```

```
 10 SPEECH INTERFERENCE , FRACTIONAL UNINTELLIGIBILITY BASED ON A-LEVEL

                          OPERATORS   PRIMARY     SECONDARY    OPERATORS
                                      DWELLING    DWELLING     + PRIMARY
                          ----------  ----------  ----------   ----------
 WEIGHTED POPULATION       .2669E+06   .6631E+06   .0           .9300E+06
 NOISE IMPACT INDEX        .1283E-02   .3188E-02   .0           .4471E-02

----------------

 11 SLEEP INTERFERENCE , DISRUPTION POTENTIAL FROM SEL

                          OPERATORS   PRIMARY     SECONDARY    OPERATORS
                                      DWELLING    DWELLING     + PRIMARY
                          ----------  ----------  ----------   ----------
 WEIGHTED POPULATION       .0          .4203E+09   .2929E+09    .4203E+09
 NOISE IMPACT INDEX        .0          .2020E+01   .1408E+01    .2020E+01

----------------

 12 SLEEP INTERFERENCE , AWAKENING POTENTIAL FROM SEL

                          OPERATORS   PRIMARY     SECONDARY    OPERATORS
                                      DWELLING    DWELLING     + PRIMARY
                          ----------  ----------  ----------   ----------
 WEIGHTED POPULATION       .0          .2787E+09   .1099E+09    .2787E+09
 NOISE IMPACT INDEX        .0          .1340E+01   .5283E+00    .1340E+01


****************

COMMAND ? LOGOFF CLR
END OF SESSION  WEDNESDAY  10/29/80  7:09:50 P.M.

EPAJHT/S2KC OFF WYLBUR 10/29/80 AT 19:09:50, 2.38 WUU
 0.14 CONNECT HRS., 0:00.13 TCB, 0 PAGE-SECONDS
 EXCPS: 11 DA, 0 MT, 386 TERM, 0 OTHER, 397 TOTAL
 CHARGES: $0.00 CONNECT, $1.33 WUU, $1.33 TOTAL
```

4. LOW LEVEL PROGRAM DESCRIPTION

This section provides a listing of all of the subroutines used in both modes of the CPNIM program. Before each subroutine is a brief written description of what goes on in the subroutine. Included will be a description of the subroutine purpose, where it is called from, which variables are passed and what, if anything is calculated in the subroutine.

Main Program for the Interactive Mode

This program reads in the files by calling READ10, READ11, READU and introduces the model on the user's terminal. It then calls OPTION which directs the program flow from that point on.

```
//EPAJHTEE JOB (S2KC,MJHT),BUFF,PASSWORD=ROBERT
/*ROUTE  PRINT HOLD
//STEP1 EXEC FTG1CL
//FORT.SYSIN DD *
C
C     MAIN ROUTINE:  CONSUMER PRODUCT NOISE IMPACT MODEL
C
C     READ FUNDAMENTAL FILE ON UNIT 10
C
      CALL READ10
C
C     READ CODE FILES ON UNIT 11
C          - OPTION CODES
C          - PARAMETER CODES
C          - PARAMETER NAMES
C          - PARAMETER SUBSCRIPTS
C          - VARIABLE NAMES
C          - VARIABLE SUBSCRIPTS
C
      CALL READ11
C
C     SET UP THE FOUR DIRECT ACCESS FILES NEEDED FOR USER INTERFACE
C         - THREE MESSAGE FILES
C              HELP
C              PROMPT
C              ERROR
C         - ONE TEMPORARY FILE USED TO STORE INPUTS AND OUTPUTS
C                FROM UP TO 20 DIFFERENT RUNS
C
      CALL READU
C
C     THIS PROGRAM ASSESSES THE HEALTH AND WELFARE
C     IMPACT OF CONSUMER PRODUCT NOISE.  THE
C     ORIGINAL VERSION WAS WRITTEN BY THE
C     NATIONAL BUREAU OF STANDARDS FOR EPA IN
C     1978.  THE MODEL WAS UPDATED AND MADE
C     INTERACTIVE IN NATURE IN 1980.
C
C     USE OF THIS PROGRAM DOES NOT REQUIRE
C     EXTENSIVE PROGRAMMING EXPERIENCE.  THE
C     MODEL IS DESIGNED TO PROMPT THE USER FOR
C     ALL INPUT DATA, AND ALLOWS MAXIMUM FLEXABILITY
C     IN PROGRAM OPERATION THROUGH EASILY UNDERSTOOD
C     DIRECTIONS.
```

```
C
C     THE OPERATOR IS PROVIDED ADDITIONAL ASSISTANCE
C     AT ANY OF THE MODEL PROMPTS BY TYPING
C          - CODE
C          - HELP
C          - OOPS
C          - SOS
C          - ?
C     TYPING 'CODE' WILL PROVIDE A LIST OF THE PARAMETER CODES.
C     TYPING 'HELP' WILL PROVIDE A MORE EXPANSIVE EXPLANATION.
C     TYPING 'OOPS' WILL RETURN THE OPERATOR TO THE BEGINNING
C     OF THE PROMPT
C     TYPING 'SOS ' WILL RETURN THE OPERATOR ONE LEVEL UP.
C     TYPING '?   ' WILL PROVIDE A LIST OF THE ASSISTANCE CODES.
C
C     DOCUMENTATION ON THE USE OF THIS PROGRAM, AND A
C     BATCH VERSION OF THE SAME PROGRAM, CAN BE
C     FOUND IN THE "USERS MANUAL AND PROGRAM GUIDE"
C     AVAILABLE THROUGH THE EPA-ONAC OFFICE.
C
C     THE MODEL CALCULATES
C          - POPULATION IMPACTS AND NOISE IMPACT INDICES
C               FRACTIONAL EXPOSURE WEIGHTING
C                    BASED ON AN LEQ OF 70
C               GENERAL ADVERSE RESPONSE
C                    CHABA OUTDOOR NOISE SOURCE
C                    CHABA INDOOR NOISE SOURCE
C                    FRACTIONAL IMPACT METHOD
C                    ALEXANDER ENERGY
C                    ALEXANDER LOUDNESS
C                    ALEXANDER SYNTHETIC
C              SPEECH INTERFERENCE, FRACTIONAL UNINTELLIGABILITY
C                    BASED ON A-LEVEL
C              SLEEP INTERFERENCE
C                    DISRUPTION POTENTIAL FROM SEL
C                    AWAKENING POTENTIAL FROM SEL
C
C          - POPULATION IMPACTS AND NOISE INDUCED PERSONAL
C             THRESHOLD SHIFTS
C               POPULATION WEIGHTED LOSS OF HEARING
C                    BASED ON CHABA CRITERION
C                    AT 4000 HZ
C
C     THIS INTRODUCES THE MODEL
C
      DO 10 I=1,3
      CALL MSG('HELP',I)
 10   CONTINUE
C
C     BEGIN
C
      CALL OPTION
C
      STOP
      END
```

Block Data for Interactive Program

This block of data initializes the program to the NBS data set for product specific variables dealing with a vacuum cleaner.

```
                BLOCK DATA
C
C      INITILIZATION OF PRODUCT SPECIFIC VARIABLES
C      USING NBS SAMPLE DATA SET
C
       COMMON /OUTPUT/ PI(4,12),IRAN,DIR(18,20),NUNIT
       COMMON /INPUTS/TITLE(18).OPTYP(6),PERC(3),
     1       ONTIM(3,3),TIM(3,3),TP9(4,3,3),
     1       RMP9(6,3,3),SPECTR(8),IDB,MIN,MAX,
     1       OPDIFF(3),DISLEV(20),IT,OPTOT
       COMMON /CONST/XDAYS(3),TPER(4),HSWT(3),NCANUM(12)
C
       DATA IRAN/0/,NUNIT/6/
       DATA XDAYS/.534, .181, .285/,
     1     TPER/120., 480., 300., 540./,
     1     HSWT/2., 2., 4./,
     1     NCANUM/1. 1, 1,  2, 2. 2,  2, 2, 2,  3, 4, 4/
C
       DATA TITLE/'NBS ','SAMP','LE D','ATA ','SET ','    ',
     1 '    ','    ','    ','    ','    ','    ','    ',
     1 '    ','    ','    ','    ','    '/,
     1 OPTYP/.1, .05, .5, .25. .1, 0.0/,
     1 PERC/.992, .992, .992/,
     1 ONTIM/25.7, 19.3, 12.8, 25.7, 19.3, 12.8, 25.7, 19.3, 12.8/,
     1 TIM/60.,45.,30.,  60.,45.,30., 60.,45.,30./,
     1 TP9/.05,.75,.2,0.0,.05,.75,.2,0.0, .05,.75,.2,0.0,
     1 .05,.75,.2,0.0,.05,.75,.2,0.0, .05,.75,.2,0.0,
     1 .05,.75,.2,0.0,.05,.75,.2,0.0, .05,.75,.2,0.0/,
     1 RMP9/.08,.5,.0,.34,.08,.00,  .11,.56,.00,.33,.00, .00,
     1      .17,.50,.00,.33,.00,.00,
     1     .08,.5,.0,.34,.08,.00.  .11,.56.00,.33,.00, .00,
     1      .17,.50,.00,.33,.00,.00,
     1     .08,.5,.0,.34,.08,.00,  .11,.56,.00,.33.00, .00,
     1      .17,.50,.00,.33,.00,.00/,
     1 SPECTR/62.2, 69.5, 71.5, 71.0, 71.1, 64.2, 56.8, 44.5/,
     1 IDB/5/,MIN/70/,MAX/95/,
     1 OPDIFF/-12.0. 0.0. 0.0/,
     1 DISLEV/.19,.33,.19,.15,.14, 15*0.0/,IT/5/,OPTOT/1./
          END
```

Main Program BACH

This is the main program executed with the instruction USE WYLIB (BATCH). It directs the flow through the acquisition of input parameters, calculation of weighted population, and the output of results to the file CPNIM.OUTPUT.

```
//EPAJHTAA JOB (S2KC,MJHT),BUFF,PASSWORD=ROBERT
/*ROUTE  PRINT HOLD
//STEP1 EXEC FTG1CL
//FORT.SYSIN DD *
C
       COMMON /DERIV / PEOPLE(3,6),SEOPLE(3,6),POPTOT,
     1                 SAV1,PROBON(3,6,4,3),SIMEIN(3,6,6,4,3),
     1                 SPECTA(8)
       COMMON /OUTPUT/ PI(4,12),IRAN,DIR(18,20),NUNIT
      COMMON /CONST/XDAYS(3),TPER(4),HSWT(3),NCANUM(12)
     1        /INPUTS/TITLE(18),OPTYP(6),PERC(3),
     1        ONTIM(3,3),TIM(3,3),TP9(4,3,3),
     1        RMP9(6,3,3),SPECTR(8),IDB,MIN,MAX,
     1        OPDIFF(3),DISLEV(20),IT,OPTOT
C
C
C
        CALL READ10
        CALL READ11
        CALL READU
C
C
1       CALL BACHIN
         CALL ERROR(&1000)
C
C      PERFORM PRELIMINARY CALCULATIONS
C
1234   CALL PRECAL(&1000)
C
C      PERFORM CALCULATIONS
C
       DO 10 ICA=1,12
       ITST=NCANUM(ICA)
            DO 30 IK=1,4
            PI(IK,ICA)=0.
            IF(IK.EQ.1.AND.OPTOT.EQ.0.)GO TO 40
            IF(IK.EQ.4.AND.OPTOT.EQ.0.)GO TO 40
            IF(ITST.EQ.4)GO TO 40
                 DO 50 M=1,6
                 DO 50 I=1,3
```

```
C
                IF(ITST.LE.2)CALL ALLEQ(ICA,IK,I,M,ITST)
                IF(ITST.EQ.3)CALL SPEECH(ICA,IK,I,M)
50              CONTINUE
40           CONTINUE
             IF(ITST.EQ.4)CALL SLEEP(ICA,IK)
30         CONTINUE
      IF(ITST.EQ.3)PI(4,ICA)=PI(1,ICA)+PI(2,ICA)
      IF(ITST.EQ.4.OR.OPTOT.EQ.0.)PI(4,ICA)=PI(2,ICA)
10         CONTINUE
C
           IRAN=IRAN+1
C
C
C     WRITE OUTPUT
C
       CALL LIST(0)
      CALL OUTOUT
C
1000      STOP
      END
```

Block Data for Batch Program

This block data, specifically for the initialization of the batch mode program, is different from the interactive mode block data in that NUNIT is set to 66.

```
C
C
C
               BLOCK DATA
C
       COMMON /DERIV / PEOPLE(3,6),SEOPLE(3,6),POPTOT,
     1                 SAV1,PROBON(3,6,4,3),SIMEIN(3,6,6,4,3),
     1                 SPECTA(8)
       COMMON /OUTPUT/ PI(4,12),IRAN,DIR(18,20),NUNIT
        COMMON/COMSG/MAPPRM(100),MAPHLP(100),MAPERR(100),
     1        LEVEL,SINGLE
     1        /INPUTS/TITLE(18),OPTYP(6),PERC(3),
     1        ONTIM(3,3),TIM(3,3),TP9(4,3,3),
     1        RMP9(6,3,3),SPECTR(8),IDB,MIN,MAX,
     1        OPDIFF(3),DISLEV(20),IT,OPTOT
       COMMON /FUNCOM/ TIMEIN(3,6,6,4,3),PEOPLM(3,6),
     1          NRP(8,6,6,3),     NRS(8,6,6,3)
      COMMON /CONST/XDAYS(3),TPER(4),HSWT(3),NCANUM(12)
       REAL NRP,NRS
C
          DATA IRAN/0/
          DATA NUNIT/66/
         DATA XDAYS/.534, .181, .285/,
     1        TPER/120., 480., 300., 540./,
     1        HSWT/2., 2., 4./,
     1        NCANUM/1, 1, 1,  2, 2, 2,  2, 2, 2,  3, 4, 4/
C
        DATA TITLE/'NBS ','SAMP','LE D','ATA ','SET ','    ',
     1   '    ','    ','    ','    ','    ','    ','    ',
     1 '    ','    ','    ','    ','    '/,
     1  OPTYP/.1, .05, .5, .25, .1, 0.0/,
     1  PERC/.992, .992, .992/,
     1  ONTIM/25.7, 19.3, 12.8, 25.7, 19.3, 12.8, 25.7, 19.3, 12.8/,
     1  TIM/60.,45.,30.,  60.,45.,30., 60.,45.,30./,
     1  TP9/.05,.75,.2,0.0,.05,.75,.2,0.0, .05,.75,.2,0.0,
     1  .05,.75,.2,0.0,.05,.75,.2,0.0, .05,.75,.2,0.0,
     1  .05,.75,.2,0.0,.05,.75,.2,0.0, .05,.75,.2,0.0/,
     1  RMP9/.08,.5,.0,.34,.08,.00,  .11,.56,.00,.33,.00, .00,
     1       .17,.50,.00,.33,.00,.00,
     1      .08,.5,.0,.34,.08,.00,  .11,.56,.00,.33,.00, .00,
     1      .17,.50,.00,.33,.00,.00,
     1     .08,.5,.0,.34,.08,.00,  .11,.56,.00,.33,.00, .00,
     1      .17,.50,.00,.33,.00,.00/,
     1  SPECTR/62.2, 69.5, 71.5, 71.0, 71.1, 64.2, 56.8, 44.5/,
     1  IDB/5/,MIN/70/,MAX/95/,
     1  OPDIFF/-12.0, 0.0, 0.0/,
     1  DISLEV/.19,.33,.19,.15,.14, 15*0.0/,IT/5/,OPTOT/1./
C
          END
```

Subroutine BACHIN

This subroutine is called by BACH to read in the product specific data from the CPNIM.INPUT file.

```
C
          SUBROUTINE BACHIN
C
       COMMON/INPUTS/TITLE(18),OPTYP(6),PERC(3),
     1      ONTIM(3,3),TIM(3,3),TP9(4,3,3),
     1      RMP9(6,3,3),SPECTR(8),IDB,MIN,MAX,
     1      OPDIFF(3),DISLEV(20),IT,OPTOT
            NAMELIST /INS/OPTYP,PERC,ONTIM,TIM,TP9,
     1        RMP9,SPECTR,IDB,MIN,MAX,OPDIFF,DISLEV
C
          READ(20,10)TITLE
10        FORMAT(18A4)
          READ(20,INS)
C
           IT=(MAX-MIN)/IDB
C
            RETURN
            END
```

Subroutine READ10

This subroutine is called by CPNIM and BACH to read in the fundamental data set from file FUND.DAT. The data is read in in NAMELIST format from NAMELIST FUNDAT.

```
      SUBROUTINE READ10
C
C      CALLED BY CPNIM
C      READS IN THE FUNDAMENTAL DATA FILE FUND.DAT
C      THIS FILE CONTAINS THE INPUT VARIABLES
C           TIMEIN(I,K,M,P,N)
C           PEOPLM(I,M)
C           NRP(S,K,J,I)
C           NRS(S,L,J,I)
C      THE FILE FUND.DAT HAS THE FUNDAMENTAL DATA NUMBERS IN
C      WYLBUR CARD FORMAT ON UNIT 10.
C
C      THE SUBSCRIPT DEFINITIONS FOR THESE VARIABLES ARE
C      CONTAINED IN THE CODDAT.DAT FILE.
C
C      THE INPUT VARIABLES ARE DEFINED BELOW
C           TIMEIN = THE LENGTH OF TIME, IN MINUTES,
C                    PEOPLE SPEND IN DIFFERENT TYPES
C                    OF ROOMS IN DIFFERENT TYPES OF HOUSES
C                    FOR DIFFERENT TIME PERIODS
C                    ON DIFFERENT TYPES OF DAYS
C
C           PEOPLM = THE TOTAL NUMBER OF PEOPLE,IN
C                    MILLIONS, THAT LIVE IN A GIVEN
C                    TYPE OF HOUSE
C
C           NRP    = THE NOISE REDUCTION WITHIN
C                    THE PRIMARY DWELLING
C
C           NRS    = THE NOISE REDUCTION BETWEEN THE
C                    PRIMARY DWELLING AND THE SECONDARY
C                    DWELLING
C
            COMMON/OUTPUT/PI(4,12),IRAN,DIR(18,20),NUNIT
       COMMON /FUNCOM/ TIMEIN(3,6,6,4,3),PEOPLM(3,6),
     1          NRP(8,6,6,3),     NRS(8,6,6,3)
       REAL NRP,NRS
C
C      FUNDAT IS THE NAME OF THE NAMELIST THAT INCLUDES
C      THE NAMES OF ALL THE VARIABLES TO BE READ FROM THE
C      FUNDAMENTAL DATA SET
```

```
C
            NAMELIST /FUNDAT/ TIMEIN,PEOPLM,NRP,NRS
C
C           READ IN THE DATA
C
            READ(10,FUNDAT)
            RETURN
                END
```

Subroutine READ11

This subroutine is called by CPNIM and BACH to read in the option codes, parameter codes, parameter names, parameter subscripts, variable names, and variable subscripts from the CODDAT.DAT file. The data is read in by Format Control.

```
      SUBROUTINE READ11
C
C     CALLED BY CPNIM
C     READS IN THE CODE FILE CODDAT.DAT
C     THE CODE FILE CONTAINS
C         - OPTION CODES
C         - PARAMETER CODES
C         - PARAMETER NAMES
C         - PARAMETER SUBSCRIPTS
C         - VARIABLE NAMES
C         - VARIABLE SUBSCRIPTS
C     THE CODDAT.DAT FILE IS STORED IN WYLBUR CARD
C     CARD FORMAT ON UNIT 11.
C
      COMMON /CODNAM/ HOUSE(6,3),ROOM(6,6),PERSON(6,6),
     1              DAY(6,3),PERIOD(6,4),BAND(6,8),
     1              CODE(6,10),OPTION(2,7),VNAMES(2,17),
     1              SUBS(9),VSIZES(5,17)
C
C     READ IN FILE
C
      READ(11,20,END=100) HOUSE,ROOM,PERSON,DAY,
     1                    PERIOD,BAND,CODE,OPTION,
     1                    VNAMES,SUBS,VSIZES
 20   FORMAT(10(/),3(6A4,/),3(/),6(6A4,/),3(/),6(6A4,/),
     1     3(/),3(6A4,/),3(/),4(6A4,/),3(/),8(6A4,/),3(/),
     1     10(6A4,/),3(/),7(2A4,/),3(/),17(2A4,/),3(/),
     1     9(A1,/),3(/),17(5A1,/))
100   RETURN
          END
```

Subroutine READU

This subroutine is called by CPNIM and BACH to set up the files for all of the messages used in the program and to set up the temporary output file for individual runs within the session.

```
      SUBROUTINE READU
C      CALLED BY CPNIM
C      SETS UP THE FOUR DIRECT ACCESS FILES USED BY THE PROGRAM
C
C
      LOGICAL SINGLE
      DIMENSION MAPPRM(100),MAPHLP(100),MAPERR(100)
      COMMON/COMSG/MAPPRM,MAPHLP,MAPERR,
     1              LEVEL,SINGLE
C
C       THERE ARE FOUR DIRECT-ACCESS FILES USED IN THIS PROGRAM
C          UNIT 30 - THE OUTPUT FROM THE MODEL IS STORED HERE
C                    THIS IS A TEMPORARY FILE
C                    EACH RECORD CONTAINS INPUT AND OUTPUT
C                    FROM A SINGLE RUN
C          UNIT 50 - PROMPT MESSAGES STORED HERE - FILE CPNIM.PROMPT
C                    INCLUDES VARIABLE DESCRIPTORS
C                    EACH RECORD CONTAINS A SINGLE LINE OF TEXT
C          UNIT 51 - HELP MESSAGES STORED HERE   - FILE CPNIM.HELP
C                    ONE LINE OF TEXT PER RECORD
C          UNIT 52 - ERROR MESSAGES STORED HERE  - FILE CPNIM.ERROR
C                    ONE LINE OF TEXT PER RECORD
C
C       THESE DEFINE FILE STATEMENTS DEFINE THE RECORD SIZE
C       AND MAXIMUM NUMBER OF RECORDS IN A FILE
C        UNIT 30 - LIMITED TO STORING DATA FROM 20 RUNS
C        UNIT 50 - LIMITED TO ABOUT 150 LINES OF PROMPTS
C        UNIT 51 - LIMITED TO ABOUT 1000 LINES OF HELP MESSAGE
C        UNIT 52 - LIMITED TO ABOUT 300 LINES OF ERROR MESSAGES
C
        DEFINE FILE 30(20,1000,L,IABC)
        DEFINE FILE 50(150,80,E,IPRM)
        DEFINE FILE 51(1000,80,E,IHLP)
        DEFINE FILE 52(300,80,E,IERR)
C
C     THIS DO LOOP READS THE THREE MAP ARRAYS NEEDED TO ACCESS
C         THE PROMPT,HELP,AND ERROR MESSAGES
C         THESE ARRAYS ARE USED IN SUBROUTINE MSG
C
C     MAPPRM(I)- POINTS TO RECORD CONTAINING LINE 1 OF PROMPT I
C     MAPHLP(I)- POINTS TO RECORD CONTAINING LINE 1 OF HELP MSG I
C     MAPERR(I)- POINTS TO RECORD CONTAINING LINE 1 OF ERROR I
C
C     THE THREE MAP ARRAYS ARE WRITTEN IN RECORDS #2-6 OF THE
C           RELATED FILE
C
```

```
         MM=1
         DO 100 II=2,6
         MSTOP=MM+19
       READ(50'II,18)(MAPPRM(KK),KK=MM,MSTOP)
       READ(51'II,18)(MAPHLP(KK),KK=MM,MSTOP)
       READ(52'II,18)(MAPERR(KK),KK=MM,MSTOP)
100     MM=MM+20
18        FORMAT(20I4)
C
       RETURN
       END
```

Subroutine OPTION

This subroutine interfaces with the user to allow the selection of the program options, and then calls the appropriate subroutine for the user response.

```
      SUBROUTINE OPTION
C
C      CALLED BY CPNIM
C      OPTION IS THE CONTROLLING USER INTERFACE SUBROUTINE
C      THE USER CAN SPECIFY THE FOLLOWING OPTIONS
C           - CHANGE
C           - INPUT
C           - LIST
C           - PRINT
C           - RUN
C           - STOP
C           - VERIFY
C
C      THESE OPTIONS ARE DEFINED BELOW:
C           - CHANGE = CHANGE PRODUCT SPECIFIC AND FUNDAMENTAL
C                      VARIABLES
C           - INPUT  = SPECIFY PRODUCT SPECIFIC VARIABLES
C           - LIST   = LIST ALL OF THE PRODUCT SPECIFIC VARIABLES
C           - PRINT  = PRINT OUTPUT ON HIGH SPEED PRINTER
C           - RUN    = RUN THE MODEL
C           - STOP   = END THE EXECUTION OF THE MODEL
C           - VERIFY = VERIFY INDIVIDUAL VARIABLE ELEMENTS
C                      OR VARIABLE ARRAYS
C
      COMMON /CODNAM/ HOUSE(6,3),ROOM(6,6),PERSON(6,6),
     1              DAY(6,3),PERIOD(6,4),BAND(6,8),
     1              CODE(6,10),OPTION(2,7),VNAMES(2,17),
     1              SUBS(9),VSIZES(5,17)
      COMMON /COMSG/ MAPPRM(100),MAPHLP(100),MAPERR(100),
     1               LEVEL,SINGLE
      REAL ALPHA(18)
C
C      ASK FOR OPTION
C
100   LEVEL = 4
      WRITE(6,200) OPTION
200   FORMAT(/,' THE OPTIONS ARE:',/,7(10X,2A4,/),/,
     1        ' ENTER OPTION:')
C
C      READ INPUT
```

```
C
        CALL ASK(VALUE,ALPHA,&100,&800,0)
C
C       IDENTIFY OPTION
C
              DO 300 I=1,7
              IF(ALPHA(1).EQ.OPTION(1,I))
     1        GO TO (400,500,600,650,700,800,900),I
300           CONTINUE
C
C       ERROR
C
        CALL MSG('ERRO',1)
        GO TO 100
C
C       SPECIFIED CHANGE OPTION
C
400     CALL CHANGE
        GO TO 100
C
C       SPECIFIED INPUT OPTION
C
500     CALL INPUT
        GO TO 100
C
C       SPECIFIED LIST OPTION
C
600     CALL LIST
        GO TO 100
C
C       SPECIFIED PRINT OPTION
C
650     CALL PRINT
        GO TO 100
C
C       SPECIFIED RUN OPTION
C
700     CALL RUN
        GO TO 100
C
C       SPECIFIED STOP OPTION
C
800     CALL FINI(&1000)
        GO TO 100
C
C       SPECIFIED VERIFY OPTION
C
900     CALL VERIFY
        GO TO 100
C
1000     RETURN
         END
```

Subroutine ASK

This subroutine is called any time a user input is requested and calls NUMCK and CHAR to ensure that the proper value has been input.

```
         SUBROUTINE ASK(VALUE,ALPHA,*,*,IFG)
C
C        CALLED BY MANY SUBROUTINES FOR USER INPUT
C        READS IN ONE LINE OF DATA
C        WRITES LINE OF CODE TO UNIT 9
C             RETURN 1 - RETURNS THE USER TO THE BEGINNING OF THE
C                        PROMPT
C             RETURN 2 - RETURNS  USER ONE LEVEL UP
C                        THE SUBROUTINE HEIRARCHY
C             RETURN   - RETURNS THE USER TO THE STATEMENT AFTER
C                        THE CALL STATEMENT.
C
         REAL ANS(72),ALPHA(18)
C
C        READ INPUT LINE
C
         READ(5,10) (ANS(II),II=1,72)
10       FORMAT(72A1)
C
C        WRITE INPUT LINE TO UNIT 9
C
         REWIND 9
         WRITE(9,10) (ANS(II),II=1,72)
C
C        CALL SUBROUTINE NUMCK TO CHECK IF INPUT WAS
C        A NUMBER
C
         IF(IFG.EQ.1) CALL NUMCK(ANS,VALUE,&50,&30)
C
C        ANSWER WAS A CHARACTER INPUT
C        READ CHARACTERS FROM UNIT 9
C
         CALL CHAR(ALPHA,&30,&40,IFG)
50       RETURN
30       RETURN 1
40       RETURN 2
         END
```

Subroutine NUMCK

This subroutine is called by ASK and checks to see if the user has input a proper number. If the value is not a valid number then the program returns to ASK and then goes to CHAR.

```
        SUBROUTINE NUMCK(ANS,VALUE,*,*)
C
C       CALLED BY ASK
C       CHECKS IF INPUT WAS A NUMBER
C       IF INPUT WAS A NUMBER IT READS
C       VALUE FROM UNIT 9
C             RETURN 1 - RETURNS WITH A VALID NUMBER
C             RETURN 2 - RETURNS WITH AN ERROR
C             RETURN   - RETURNS TO CHECK FOR ALPHANUMERIC INPUT
C
        INTEGER ANS(80), DJ(14)
        LOGICAL IDPOIN,INUMB
        DATA DJ/ '0','1','2','3','4','5','6','7',
     1          '8','9','+','-','.',' '/
C
C       INITIALIZE FLAGS:
C           IDPOIN = FLAG FOR DECIMAL POINT CHARACTER '.'
C           INUMB  = FLAG FOR DIGITS '0' TO '9'
C
       IDPOIN=.FALSE.
       INUMB=.FALSE.
C
C      CHECK TO SEE IF FIRST CHARACTER IS A '+' OR '-'
C
       ISTART=1
       IF(ANS(1).EQ.DJ(11).OR.ANS(1).EQ.DJ(12))ISTART=2
C
C      LOOK AT ALL CHARACTERS ON THE INPUT STRING
C
       DO 100 I=ISTART,72
C
C      CHECK FOR DIGITS '0' TO '9'
C
           DO 50 JJJ=1,10
           IF(ANS(I).EQ.DJ(JJJ))GO TO 60
50         CONTINUE
70     IF(ANS(I).EQ.DJ(14).AND.INUMB)GO TO 300
       IF(ANS(I).EQ.DJ(14).AND..NOT.INUMB)GO TO 500
       IF(ANS(I).EQ.DJ(13).AND..NOT.IDPOIN)GO TO 90
       IF(ANS(I).EQ.DJ(13).AND.IDPOIN)GO TO 600
```

```
C
C       FOUND AN ALPHANUMERIC CHARACTER
C
        RETURN
C
C       FOUND A NUMBER
C
60      INUMB=.TRUE.
        GO TO 100
C
C       FOUND A DECIMAL POINT
C
90      IDPOIN=.TRUE.
100     CONTINUE
C
C       YOU GOT A NUMBER 72 CHARACTERS LONG
C
        GO TO 450
C
C       CHECK IF THE NUMBER IS FOLLOWED BY BLANKS
C
300          DO 400 III=1,72
             IF(ANS(III).NE.DJ(14))GO TO 700
400          CONTINUE
C
C       YOU GOT A GOOD NUMBER FOLLOWED BY BLANKS
C
450     READ(9,*)VALUE
        RETURN 1
C
C       *** ERROR ***
C       NO DIGIT FOUND BEFORE BLANK ENCOUNTERED
C
500     CALL MSG('ERRO',3)
        RETURN 2
C
C       *** ERROR ***
C       MULTIPLE DECIMAL POINTS IN ENTERED NUMBER
C
600     CALL MSG('ERRO',4)
        RETURN 2
C
C       *** ERROR ***
C       VALID NUMBER NOT FOLLOWED BY BLANKS
C
700     CALL MSG('ERRO',5)
        RETURN 2
        END
```

Subroutine CHAR

This subroutine is called by ASK and checks to see if the user has requested one of the assistance codes. If the user input is:

CODE - the subroutine calls CODE
HELP - the subroutine calls MSG
OOPS - returns to prompt message
SOS - returns to the next subroutine level
? - the subroutine calls MSG for help message #2

If an illegal value is entered then the subroutine returns to the user's terminal with THIS IS NOT A NUMBER.

```
      SUBROUTINE CHAR(ALPHA,*,*,IFG)
        DIMENSION TEMP(18)
C
C     CALLED BY ASK
C     CAN BE CALLED ANYTIME YOU NEED A SHORT,
C     NON NUMERIC RESPONSE
C             RETURN 1 - RETURNS THE USER TO THE BEGINNING OF THE
C                        PROMPT
C             RETURN 2 - RETURNS  USER ONE LEVEL UP
C                        THE SUBROUTINE HEIRARCHY
C             RETURN   - RETURNS THE USER TO THE STATEMENT AFTER
C                        THE CALL STATEMENT.
C
      COMMON /COMSG/ MAPPRM(100),MAPHLP(100),MAPERR(100),
     1               LEVEL,SINGLE
      REAL   AID(5),ALPHA(18)
      DATA AID/'CODE','HELP','OOPS','SOS ','?   '/
C
C     READ FIRST FOUR ALPHNUMERIC CHARACTERS
C     FROM UNIT 9
C
      REWIND 9
      READ(9,10)TEMP
10    FORMAT(18A4)
C
C     CHECK TO SEE IF ANSWER EQUALS
C          - CODE
C          - HELP
C          - OOPS
C          - SOS
C          - ?
C
      DO 11 I=1,5
           IF(TEMP(1).EQ.AID(I))  GO TO (20,30,40,50,60).I
11         CONTINUE
```

```
C
C      RETURN WITH THE USER SUPPLIED ANSWER
C
         DO 12 III=1,18
12       ALPHA(III)=TEMP(III)
C
       IF(IFG.EQ.0) RETURN
       CALL MSG('ERRO',2)
       RETURN 1
C
C      A LIST OF THE PARAMETER CODES IS NEEDED
C
20     CALL CODE
       RETURN 1
C
C      HELP NEEDED
C
30     CALL MSG('HELP',LEVEL)
       RETURN 1
C
C
C      RETURN TO THE BEGINNING OF THE PROMPT
C
40     RETURN 1
C
C      RETURN ONE LEVEL UP
C
50     RETURN 2
C
C      GET ASSISTANCE CODES
C
60     CALL MSG('HELP',2)
       RETURN 1
        END
```

Subroutine CODE

This subroutine is called by CHAR and writes the parameter codes and names to the user's terminal.

```
               SUBROUTINE CODE
     C       CALLED BY SUBROUTINE CHAR
     C      LISTS THE PARAMETER SUBSCRIPT NAMES AND THEIR
     C       RESPECTIVE CODES
     C
                     COMMON /CODNAM/HOUSE(6,3),ROOM(6,6),PERSON(6,6),
          1           DAY(6,3),PERIOD(6,4),BAND(6,8),
          1           CODE(6,10),OPTION(2,7),VNAMES(2,17)
          1        ,SUBS(9),VSIZES(5,17)
                COMMON/OUTPUT/PI(4,12),IRAN,DIR(18,20),NUNIT
     C       WRITE PARAMETER CODES AND NAMES
     C
               WRITE(NUNIT,10) (CODE(I,1),I=1,6),HOUSE,
          1                ((CODE(I,J),I=1,6),J=2,5),ROOM,
          1                (CODE(I,6),I=1,6),PERSON,
          1                (CODE(I,7),I=1,6),DAY,
          1                (CODE(I,8),I=1,6),PERIOD,
          1                (CODE(I,9),I=1,6),BAND
     10      FORMAT(/,1X,6A4,/,3(6X,6A4,/),/,2(1X,6A4,/),2(6A4,/),6(6X,6A4,/
),
          1          /,1X,6A4,/,6(6X,6A4,/),/,1X,6A4,/,3(6X,6A4,/),
          1          /,1X,6A4,/,4(5X,6A4,/),/,1X,6A4,/,8(5X,6A4,/))
                RETURN
                END
```

Subroutine MSG

This subroutine is called at any time in the program where a prompt, help, or error message is needed.

```
        SUBROUTINE MSG(TYPE,LEV)
       COMMON/COMSG/MAPPRM(100),MAPHLP(100),MAPERR(100),
     1    LEVEL,SINGLE
       COMMON /OUTPUT/ PI(4,12),IRAN,DIR(18,20),NUNIT
         INTEGER POINT
         LOGICAL SINGLE
        DIMENSION LINE(18),MAPH(100)
        DATA STOPM/'END '/
         DATA LANK/'    '/
        DATA XHLP/'HELP'/
        DATA XERR/'ERRO'/
        DATA XPRM/'PROM'/
C
        IF(TYPE.EQ.XHLP)GO TO 10
        IF(TYPE.EQ.XERR)GO TO 20
        IF(TYPE.EQ.XPRM)GO TO 30
C         THIS IS A HELP REQUEST
C
 10      IUNIT=51
         NOWREC=MAPHLP(LEV)
         GO TO 500
C
C   THIS IS A ERROR MESSAGE REQUEST
20       IUNIT=52
         WRITE(NUNIT,21)LEV
21       FORMAT(' *** ERROR NUMBER ',I4,' ***')
C
         NOWREC=MAPERR(LEV)
         GO TO 500
C
C
C    THIS IS A PROMPT REQUEST
 30      IUNIT=50
         NOWREC=MAPPRM(LEV)
C
C
C
C
500    IF(NOWREC.NE.0)GO TO 100
C
C IF HERE MEANS  THERE IS A PROGRAMMING ERROR
C
       WRITE(NUNIT,101)
101    FORMAT(' THERE IS A PROGRAMMING ERROR')
```

```
               RETURN
C
100         READ(IUNIT'NOWREC,151)(LINE(II),II=1,18),POINT
151          FORMAT(18A4,4X,I4)
            WRITE(NUNIT,152)(LINE(II),II=1,18)
152         FORMAT(1X,18A4)
C
            IF (POINT.EQ.0)RETURN
            NOWREC=POINT
            GOTO 100
              END
```

Subroutine PRINT

This subroutine is called by OPTION and allows the user to print outputs and inputs from any of the runs executed during this session. The output is directed either to the user's terminal or to the file CPNIM.OUTPUT for listing after the session is closed.

```
C
C
          SUBROUTINE PRINT
            LOGICAL INTOO,MI2RUN
         COMMON /OUTPUT/ PI(4.12),IRAN,DIR(18,20),NUNIT
          COMMON/COMSG/MAPPRM(100),MAPHLP(100),MAPERR(100),
     1        LEVEL,SINGLE
     1        /INPUTS/TITLE(18).OPTYP(6),PERC(3),
     1        ONTIM(3,3),TIM(3,3),TP9(4,3,3),
     1        RMP9(6,3,3),SPECTR(8),IDB,MIN,MAX,
     1        OPDIFF(3),DISLEV(20),IT
C
            DIMENSION MI2RUN(20),ARRAY(18)
            DATA YES/'YES'/
            DATA FFFFF/'F'/
1           NUNIT=6
            INTOO=.FALSE.
            IF(IRAN.NE.0)GO TO 5
            CALL MSG('ERRO',29)
C       NOTHING TO TYPE OUT YET
             RETURN
C
5           WRITE(NUNIT,10)IRAN
10          FORMAT(' WHICH ONE OF THESE',I3,' RUNS DO YOU WANT ',
     1         'TO LIST?')
C
            DO 1000 III=1,IRAN
            WRITE(NUNIT,20)III.(DIR(KKK,III),KKK=1,18)
20          FORMAT(I3,1X,18A4)
1000         CONTINUE
C
           WRITE(NUNIT.21)
21         FORMAT(1X)
C
           CALL GETLST(MI2RUN,&1,&9000)
C
39          CALL MSG('PROM',28)
C
            LEVEL=41
           CALL ASK(DUMMY.ARRAY.&39.&9000,0)
```

```
        IF(ARRAY(1).EQ.FFFFF)NUNIT=66
        LASNUN=NUNIT
        NUNIT=6
        IF(ARRAY(1).EQ.FFFFF)CALL MSG('PROM',27)
        NUNIT=LASNUN
C
        DO 50 III=1,IRAN
        IF(MI2RUN(III))GO TO 29
50      CONTINUE
C
        CALL MSG('ERRO',38)
        GO TO 1
C
C
29       WRITE(6,30)
30      FORMAT(/,' DO YOU WANT TO LIST THE INPUTS FOR ',
     1          'EACH CASE TOO?',/,'  TYPE YES OR NO')
         LEVEL=41
       CALL ASK(DUMMY,ARRAY,&29,&9000,0)
        IF(ARRAY(1).EQ.YES)INTOO=.TRUE.
C
        DO 300 III=1,IRAN
        IF(.NOT.MI2RUN(III))GO TO 300
        CALL GET(III)
         WRITE(NUNIT,250)III
250      FORMAT(/,25(1H*),//,' RUN #',I3,/)
        IF(INTOO)CALL LIST(0)
        CALL OUTOUT
300     CONTINUE
        WRITE(NUNIT,251)
251     FORMAT(/,25(1H*),//)
C
9000     NUNIT=6
       RETURN
        END
```

Subroutine GETLST

This subroutine is called by PRINT and allows the user to specify several runs to be output by inputing several numbers separated by commas.

```
                SUBROUTINE GETLST(MI2RUN,*,*)
              COMMON /OUTPUT/ PI(4,12),IRAN,DIR(18,20),NUNIT
                LOGICAL MI2RUN(20)
                 INTEGER CRRAY(20)
      C
                DIMENSION DJ(12),ARRAY(18),BRRAY(72)
                DATA DJ/'1','2','3','4','5','6','7','8','9','0',
           1               ',',' '/
      C
      1            DO 110 III=1,20
                   CRRAY(III)=0
      110          MI2RUN(III)=.FALSE.
      C
                 CALL ASK(DUMMY,ARRAY,&9000,&9000,0)
      C
                 REWIND 9
                 READ(9,10)BRRAY
      10         FORMAT(72A1)
      C
                 DO 50 III=1,72
                 DO 40 JJJ=1,11
                 IF(DJ(JJJ).EQ.BRRAY(III))GO TO 50
      40         CONTINUE
                     IF(DJ(12).EQ.BRRAY(III))GOTO 49
                 CALL MSG('ERRO',30)
                 RETURN 1
      49             BRRAY(III)=DJ(11)
      50         CONTINUE
      C
                   REWIND 9
                  WRITE(9,51)BRRAY
      51         FORMAT(72A1)
      C
                 REWIND 9
                  READ(9,*)CRRAY
      C
      C
      C
                   IF(CRRAY(III).EQ.0)GO TO 200
                  DO 200 III=1,20
                  IF(CRRAY(III).GE.0.AND.CRRAY(III).LE.IRAN)GO TO 190
                    CALL MSG('ERRO',34)
```

```
             GO TO 1
190              MI2RUN(CRRAY(III))=.TRUE.
200        CONTINUE
C
           RETURN
9000        RETURN 2
            END
```

Subroutine GET

This subroutine is called by PRINT to read input and output values for a particular run from the temporary output file.

```
C
         SUBROUTINE GET(IRUND)
        COMMON /OUTPUT/ PI(4,12),IRAN,DIR(18,20),NUNIT
         COMMON/COMSG/MAPPRM(100),MAPHLP(100),MAPERR(100),
      1         LEVEL,SINGLE
      1         /INPUTS/TITLE(18),OPTYP(6),PERC(3),
      1         ONTIM(3,3),TIM(3,3),TP9(4,3,3),
      1         RMP9(6,3,3),SPECTR(8),IDB,MIN,MAX,
      1         OPDIFF(3),DISLEV(20),IT
          READ(30'IRUND)TITLE,OPTYP,PERC,
      1  ONTIM,TIM,TP9,
      1  RMP9,SPECT.IDB,MIN,MAX,
      1  OPDIFF,DISLEV,IT.
      1  PI,POPTOT,SAV1
C
           RETURN
           END
```

Subroutine OUTOUT

This subroutine is called by RUN and PRINT and controls the printing of the outputs on the user's terminal or to the file CPNIM.OUTPUT.

```
             SUBROUTINE OUTOUT
           COMMON /DERIV / PEOPLE(3,6),SEOPLE(3,6),POPTOT,
         1                    SAV1,PROBON(3,6,4,3),SIMEIN(3,6,6,4,3),
         1                    SPECTA(8)
           COMMON /OUTPUT/ PI(4,12),IRAN,DIR(18,20),NUNIT
            COMMON/COMSG/MAPPRM(100),MAPHLP(100),MAPERR(100),
         1        LEVEL,SINGLE
         1        /INPUTS/TITLE(18),OPTYP(6),PERC(3),
         1        ONTIM(3,3),TIM(3,3),TP9(4,3,3),
         1        RMP9(6,3,3),SPECTR(8),IDB,MIN,MAX,
         1        OPDIFF(3),DISLEV(20),IT
            COMMON /FUNCOM/ TIMEIN(3,6,6,4,3),PEOPLM(3,6),
         1           NRP(8,6,6,3),     NRS(8,6,6,3)
            REAL NRP,NRS
      C
               INTEGER OUTPRO,OUTITL
             DIMENSION OUTPRO(12),OUTITL(12),WNAME(6,3),
         1               PIOVA(4)
      C
              DATA OUTPRO/51,52,53,54,55,56,57,58,59,60,61,62/
              DATA OUTITL/1,2,3,1,1,1,1,1,1,1,1,1/
      C
             DATA WNAME/'NOIS','E IM','PACT',' IND','EX  ','    ',
         1                'AVER','AGE ',' NIP','TS  ','    ','    ',
         1                'AVER','AGE ','NIPT','S AT',' 400','0HZ '/
      C
      C
      10        FORMAT(//,16(1H*),/)
      9       FORMAT(16(1H-),/)
      8       FORMAT(1X)
      C
                 WRITE(NUNIT,10)
                 WRITE(NUNIT,20)TITLE,SAV1,POPTOT
      20         FORMAT(////,1X,18A4,//
         1        ' EXPOSED PERSONS = ',E9.4,/,
         1        ' TOTAL POPULATION= ',E9.4,//)
      C
               DO 900 ICA=1,12
                    WRITE(NUNIT,9)
                CALL MSG('PROM',OUTPRO(ICA))
                WRITE(NUNIT,8)
                CALL MSG('PROM',50)
               DO 600 IK=1,4
             PIOVA(IK)=PI(IK,ICA)/SAV1
600           CONTINUE
           WRITE(NUNIT,11)(PI(IK,ICA),IK=1,4),
     1     (WNAME(KKK,OUTITL(ICA)),KKK=1,6),(PIOVA(IK),IK=1,4)
C
11          FORMAT(' WEIGHTED POPULATION     ',1X,4(E9.4,3X),/,
     1       1X,6A4,1X,4(E9.4,3X),/)
C
900           CONTINUE
          WRITE(NUNIT,10)
          RETURN
           END
```

Subroutine RUN

This subroutine is called by OPTION and controls the calculations of the impacted populations for the twelve weighting functions.

```
          SUBROUTINE RUN
           DIMENSION ARRAY(18)
             DATA ANONO/'NO'/
C         CALLED BY OPTION
C
          COMMON /DERIV / PEOPLE(3,6),SEOPLE(3,6),POPTOT,
     1                    SAV1,PRODON(3,6,4,3),SIMEIN(3,6,6,4,3),
     1                    SPECTA(8)
          COMMON /OUTPUT/ PI(4,12),IRAN,DIR(18,20),NUNIT
         COMMON /CONST/XDAYS(3),TPER(4),HSUT(3),NCANUM(12)
     1          /INPUTS/TITLE(18),OPTYP(6),PERC(3),
     1          ONTIM(3,3),TIM(3,3),TP9(4,3,3),
     1          RMP9(6,3,3),SPECTR(8),IDB,MIN,MAX,
     1          OPDIFF(3),DISLEV(20),IT,OPTOT
C
C
            CALL ERROR(&1000)
C
C
C         PERFORM PRELIMINARY CALCULATIONS
C
1234      CALL PRECAL(&1000)
C
C         PERFORM CALCULATIONS
C
          DO 10 ICA=1,12
          ITST=NCANUM(ICA)
               DO 30 IK=1,4
               PI(IK,ICA)=0.
               IF(IK.EQ.1.AND.OPTOT.EQ.0.)GO TO 40
               IF(IK.EQ.4.AND.OPTOT.EQ.0.)GO TO 40
               IF(ITST.EQ.4)GO TO 40
                     DO 50 M=1,6
                     DO 50 I=1,3
C THIS IS A COMMENTED OUT WRITE STATEMENT
COMMENT        WRITE(NUNIT,269)ICA,IK,I,M,ITST
269            FORMAT( ' ICA      IK       I        M         ITST',/,
     1    5I9)
C
```

```
                 IF(ITST.LE.2)CALL ALLEQ(ICA.IK,I,M,ITST)
                 IF(ITST.EQ.3)CALL SPEECH(ICA,IK,I,M)
50               CONTINUE
40            CONTINUE
              IF(ITST.EQ.4)CALL SLEEP(ICA,IK)
30          CONTINUE
       IF(ITST.EQ.3)PI(4,ICA)=PI(1,ICA)+PI(2,ICA)
       IF(ITST.EQ.4.OR.OPTOT.EQ.0.)PI(4,ICA)=PI(2,ICA)
10          CONTINUE
C
90      CALL STORE
C
533      CALL MSG('PROM',38)
         LEVEL=41
          CALL ASK(DUMMY,ARRAY,&533,&1000,0)
         IF(ARRAY(1).EQ.ANONO)GO TO 1000
C
C      WRITE OUTPUT
C
       CALL OUTOUT
C
1000       RETURN
       END
```

Subroutine ERROR

This subroutine is called by RUN and checks whether the input values follow their given constraints.

```
        SUBROUTINE ERROR(*)
C
C       CALLED BY RUN
C       CHECKS WHETHER VARIABLE VALUES
C       FOLLOW THEIR GIVEN CONSTRAINTS
C
        LOGICAL RTRN
        INTEGER P,S,T
        COMMON/COMSG/MAPPRM(100),MAPHLP(100),MAPERR(100),
     1        LEVEL,SINGLE
        COMMON/CODNAM/HOUSE(6,3),ROOM(6,6),PERSON(6,6),
     1        DAY(6,3),PERIOD(6,4),BAND(6,8),
     1        CODE(6,10),OPTION(2,7),VNAMES(2,17),
     1               SUBS(9),VSIZES(5,17)
        COMMON/INPUTS/TITLE(18),OPTYP(6),PERC(3),
     1        ONTIM(3,3),TIM(3,3),TP9(4,3,3),
     1        RMP9(6,3,3),SPECTR(8),IDB,MIN,MAX,
     1        OPDIFF(3),DISLEV(20),IT,OPTOT
C
C
        RTRN=.TRUE.
        OPTOT=0.
             DO 10 M=1,6
             IF((OPTYP(M).GT.1.).OR.(OPTYP(M).LT.0.))GO TO 110
10           OPTOT=OPTOT+OPTYP(M)
             IF((OPTOT.LE.1.01).AND.(OPTOT.GE.0.99))OPTOT=1.
             IF((OPTOT.NE.1.).AND.(OPTOT.NE.0.))GO TO 115
C
C
21           DO 20 I=1,3
             IF((PERC(I).GT.1.).OR.(PERC(I).LT.0.))GO TO 120
20           CONTINUE
C
C
31           DO 35 I=1,3
             DO 35 N=1,3
             SUM=0.
                  DO 30 P=1,4
                  IF((TP9(P,I,N).GT.1.).OR.(TP9(P,I,N).LT.0.))
     1            GO TO 130
                  SUM=SUM +TP9(P,I,N)
30                CONTINUE
```

```
             IF((SUM.GT.1.01).OR.(SUM.LT.0.99))GO TO 135
35          CONTINUE
C
C
41           DO 45 I=1,3
             DO 45 N=1,3
             SUM=0.
                  DO 40 J=1,6
                  IF((RMP9(J,I,N).GT.1.).OR.(RMP9(J,I,N).LT.0.))
     1            GO TO 140
                  SUM=SUM +RMP9(J,I,N)
40                CONTINUE
             IF((SUM.GT.1.01).OR.(SUM.LT.0.99))GO TO 145
45          CONTINUE
C
C
51      SUM=0.
             DO 50 T=1,IT
             SUM=SUM +DISLEV(T)
             IF((DISLEV(T).GT.1.).OR.(DISLEV(T).LT.0.))GO TO 150
50           CONTINUE
             IF((SUM.GT.1.01).OR.(SUM.LT.0.99))GO TO 155
60           CONTINUE
C
             IF(RTRN)RETURN
            CALL MSG('ERRO',24)
             RETURN 1
C
C
C       ***ERROR***
C       OPTYP MUST BE A POSITIVE FRACTION
C
110     CALL MSG('ERRO',16)
        RTRN=.FALSE.
        GO TO 21
C
C
C      ***ERROR****
C
C
115     CALL MSG('ERRO',17)
        RTRN=.FALSE.
        GO TO 21
C
C
C      ***ERROR***
C       PERC MUST BE A POSITIVE FRACTION
C
120     CALL MSG('ERRO',18)
        RTRN=.FALSE.
        GO TO 31
C
C
C       ***ERROR***
C       TP9 MUST BE A POSITIVE FRACTION
C
130     CALL MSG('ERRO',19)
        RTRN=.FALSE.
        GO TO 41
C
C
C       ****ERROR***
C
135     CALL MSG('ERRO',20)
        RTRN=.FALSE.
        GO TO 41
C
C
C       ***ERROR***
C       RMP9 MUST BE A POSITIVE FRACTION
C
140     CALL MSG('ERRO',21)
        RTRN=.FALSE.
        GO TO 51
C
C
C       ***ERROR***
C
145     CALL MSG('ERRO',22)
        RTRN=.FALSE.
        GO TO 51
C
C
C      ***ERROR***
C       DISLEV MUST BE A POSITIVE FRACTION
C
150     CALL MSG('ERRO',23)
        RTRN=.FALSE.
        GO TO 60
C
C
C      ***ERROR****
C
C
155     CALL MSG('ERRO',25)
        RTRN=.FALSE.
        GO TO 60
C
       END
```

Subroutine PRECAL

This subroutine is called from the main program in batch or the RUN option subroutine in the interactive program to calculate the values needed for the impact calculations which are common to all the weighting functions. Calculated values are:

POPTOT - total United States population.

SEOPLE - number of Exposed Persons.

PROBON - the probability that the product is on in a given dwelling, in a given room, in a given time period, on a given day type.

OPTIM - the time each person type spends operating the product.

SIMEIN - the time each person type is in a given room, excluding the time as an operator.

SPECTR - the normalized sound power spectrum.

```
        SUBROUTINE PRECAL(*)
C
C     CALLED BY SUBROUTINE RUN
C     CALCULATES VARIABLES FROM INPUT VALUES NEEDED
C     FOR THE MODEL RUN
C
        COMMON /DERIV / PEOPLE(3,6),SEOPLE(3,6),POPTOT,
     1                  SAV1,PROBON(3,6,4,3),SIMEIN(3,6,6,4,3),
     1                  SPECTA(8)
        COMMON /OUTPUT/ PI(4,12)
         COMMON/COMSG/MAPPRM(100),MAPHLP(100),MAPERR(100),
     1        LEVEL,SINGLE
     1        /INPUTS/TITLE(18),OPTYP(6),PERC(3),
     1        ONTIM(3,3),TIM(3,3),TP9(4,3,3),
     1        RMP9(6,3,3),SPECTR(8),IDB,MIN,MAX,
     1        OPDIFF(3),DISLEV(20),IT,OPTOT
        COMMON /FUNCOM/ TIMEIN(3,6,6,4,3),PEOPLM(3,6),
     1          NRP(8,6,6,3),     NRS(8,6,6,3)
       COMMON /CONST/XDAYS(3),TPER(4),HSWT(3),NCANUM(12)
        REAL NRP,NRS
        INTEGER P,S
C
C       CALCULATE PEOPLE(I,M),SAV1,SEOPLE(I,M),AND POPTOT
```

```
C      PEOPLE = THE NUMBER OF PEOPLE LIVING IN A GIVEN HOUSE
C                 OF A GIVEN PERSON TYPE
C      SAV1= THE TOTAL NUMBER OF POTENTIALLY
C               EXPOSED PERSONS IN THE PRIMARY DWELLING
C      SEOPLE = THE LENGTH OF TIME IN MINUTES
C                   DIFFERENT TYPES OF PEOPLE SPEND IN
C                   DIFFERENT TYPES OF HOUSES FOR
C                   DIFFERENT TIME PERIODS ON DIFFERENT
C                   TYPES OF DAYS EXCLUDING OPERATOR
C                   TIME.
C      POPTOT = THE TOTAL POPULATION
C
      POPTOT=0.
      SAV1=0.
       DO 10 M=1,6
            DO 10 I=1,3
            PEOPLE(I,M)=PEOPLM(I,M)*1.0E06
            SEOPLE(I,M)=PEOPLE(I,M)*PERC(I)
            POPTOT      =POPTOT+PEOPLE(I,M)
10          SAV1        =SAV1+SEOPLE(I,M)
C
C      CALCULATE PROBON
C
      DO 20 N=1,3
      DO 20 I=1,3
      DO 20 J=1,6
      DO 20 P=1,4
      PROBON(I,J,P,N)=ONTIM(I,N)*TP9(P,I,N)*RMP9(J,I,N)/TPER(P)
C
C     TIMIN AS INPUT INCLUDES OPERATOR TIME
C     SUBRACT OF OPERATOR TIME FROM TIMEIN
C     TO CALCULATE SIMEIN
C
            DO 20 M=1,6
            OPTIM=TPER(P)*PROBON(I,J,P,N)*OPTYP(M)
            SIMEIN(I,J,M,P,N)=TIMEIN(I,J,M,P,N)-OPTIM
C
C     ALLOW FOR SMALL DISCREPANCIES IN PROBON VS TIMIN
C
      IF(SIMEIN(I,J,M,P,N).GT.-.2)GO TO 20
C     ERROR
C
      CALL MSG('ERRO',12)
      WRITE(NUNIT,999) I,J,M,P,N
999   FORMAT(' SIMEIN(',4(I2,','),I2,')')
      RETURN 1
20    CONTINUE
```

Subroutine ALLEQ

This subroutine is called by RUN and calculates the impacted populations for the weighting functions which require L_{eq} or L_{dn} type of noise level data.

```
       SUBROUTINE ALLEQ(ICA,IK,I,M,ITST)
C
        COMMON /DERIV / PEOPLE(3,6),SEOPLE(3,6),POPTOT,
      1                  SAV1,PROBON(3,6,4,3),SIMEIN(3,6,6,4,3),
      1                  SPECTA(8)
        COMMON /OUTPUT/ PI(4,12).IRAN,DIR(18,20)
      1         /INPUTS/TITLE(18),OPTYP(6),PERC(3),
      1         ONTIM(3,3),TIM(3,3),TP9(4,3,3),
      1         RMP9(6,3,3),SPECTR(8),IDB,MIN,MAX,
      1         OPDIFF(3),DISLEV(20),IT,OPTOT
        COMMON /FUNCOM/ TIMEIN(3,6,6,4,3),PEOPLM(3,6),
      1           NRP(8,6,6,3),     NRS(8,6,6,3)
       COMMON /CONST/XDAYS(3),TPER(4),HSWT(3),NCANUM(12)
       INTEGER P,S,T
        REAL NRP,NRS,XLEQ(8,3,6,4),A1(8),A6(8),A2(8)
       IF(ICA.GT.1.AND.ICA.LT.4)GO TO 730
       IF(ICA.GT.4.AND.ICA.LT.10)GO TO 730
C
       IF(IK.GT.3)GO TO 730
       IF(IK.NE.1)GO TO 24
       DO 23 S=1,8
C  THIS NEXT STATEMENT ASSUMES REAL NUMBERS OVERFLOW AT 10**-79
23     XLEQ(S,I,M,1)=-78.
       IF(OPTYP(M).EQ.0.)RETURN
24     CONTINUE
       DO 27 S=1,8
       XLEQ(S,I,M,IK)=0.
       IF(IK.NE.1)GO TO 705
C
C
       X1=0
             DO 718 N=1,3
718          X1=X1+ONTIM(I,N)*XDAYS(N)*OPTYP(M)
        IF(X1.EQ.0.0) GO TO 719
       XLEQ(S,I,M,1)=SPECTA(S)+10.*ALOG10(X1/1440.)
719     IF(X1.EQ.0.0)XLEQ(S,I,M,1)=-78
       GO TO 27
705    CONTINUE
             DO 39 N=1,3
             SAV2=0
```

```
                 DO 26  J=1,6
                 DO 26  K=1,6
                 DO 26  P=1,4
                 AA=SIMEIN(I,K,M,P,N)
                 BB=PROBON(I,J,P,N)
                 IF(IK.EQ.2)X1=SPECTA(S)-NRP(S,K,J,I)
                 IF(IK.EQ.3)X1=SPECTA(S)-NRS(S,K,J,I)
                 IF(ITST.EQ.2.AND.P.EQ.4)X1=X1+10.
                 CC=10.**(X1/10.)
                 SAV2=SAV2+AA*BB*CC
   26            CONTINUE
   39        XLEQ(S,I,M,IK)=XLEQ(S,I,M,IK)+XDAYS(N)*SAV2
          IF(XLEQ(S,I,M,IK).EQ.0.) GO TO 900
       XLEQ(S,I,M,IK)=10.*ALOG10(XLEQ(S,I,M,IK)/1440.)
  900     IF(XLEQ(S,I,M,IK).EQ.0.)XLEQ(S,I,M,IK)=-78.
   27  CONTINUE
  730  CONTINUE
C
C
C
       SUM=0.
       DO 717 ITI=1,IT
       IF(IK.EQ.4)GO TO 825
            DO 716 S=1,8
  716       A1(S)=XLEQ(S,I,M,IK)+MIN+(ITI-1)*IDB+OPDIFF(IK)
       GO TO 826
  825       DO 827 S=1,8
            AA1=XLEQ(S,I,M,1)+MIN+(ITI-1)*IDB+OPDIFF(1)
            AA2=XLEQ(S,I,M,2)+MIN+(ITI-1)*IDB
  827       A1(S)=10.*ALOG10(10.**(AA1/10.)+10.**(AA2/10.))
  826  CALL WWIGHT(ICA,A1,A)
       CALL WE(W,A,ICA)
       A=SEOPLE(I,M)
  717  SUM=SUM+A*W*DISLEV(ITI)
       B=1.
       IF(IK.EQ.3)B=HSUT(I)
       PI(IK,ICA)=PI(IK,ICA)+B*SUM
       RETURN
       END
```

Subroutine SPEECH

This subroutine is called by RUN and calculates the impacted populations for the speech interference weighting function.

```
       SUBROUTINE SPEECH(ICA,IK,I,M)
C
       COMMON /DERIV / PEOPLE(3,6),SEOPLE(3,6),POPTOT,
     1                 SAV1,PROBON(3,6,4,3),SIMEIN(3,6,6,4,3),
     1                 SPECTA(8)
       COMMON /OUTPUT/ PI(4,12),IRAN,DIR(18,20)
     1        /INPUTS/TITLE(18),OPTYP(6),PERC(3),
     1        ONTIM(3,3),TIM(3,3),TP9(4,3,3),
     1        RMP9(6,3,3),SPECTR(8),IDB,MIN,MAX,
     1        OPDIFF(3),DISLEV(20),IT,OPTOT
       COMMON /FUNCOM/ TIMEIN(3,6,6,4,3),PEOPLM(3,6),
     1          NRP(8,6,6,3),     NRS(8,6,6,3)
      COMMON /CONST/XDAYS(3),TPER(4),HSWT(3),NCANUM(12)
      INTEGER P,S,T
       REAL NRP,NRS,XLEQ(8,3,6,4),A1(8),A6(8),A2(2),
     1      LEV(16),U(16),XT(2)
      DATA LEV/50.,55.,60.,64.,65.,66.,67.,68.,69.,70.,
     1         71.,72.,73.,74.,75.,76./
      DATA U  /0.,.01,.025,.05,.08,.1,.13,.18,.265,
     1         .37,.51,.68,.81,.905,.99,1.0/
      IF(IK.GT.2)RETURN
      IF(IK.EQ.2)GO TO 705
      IF(OPTYP(M).EQ.0.)RETURN
C
      A2(1)=0.
      DO 38 T=1,IT
      CC=0
      X1=MIN+(T-1)*IDB+OPDIFF(1)
      IF(X1.LT.50.)GO TO 38
      IF(X1.GT.76.)X1=75.9
      CC=AITINT(LEV,U,16,X1,2,XT)
707   A2(1)=A2(1)+CC*DISLEV(T)
38    CONTINUE
      X1=0.
      DO 718 N=1,3
718   X1=X1+ONTIM(I,N)*XDAYS(N)*OPTYP(M)
      PI(1,ICA)=PI(1,ICA)+A2(1)*X1*SEOPLE(I,M)/900.
      RETURN
705   CONTINUE
      DO 36 T=1,IT
      A2(2)=0.
           DO 39 N=1,3
           SAV2=0.
                DO 26 J=1,6
                DO 26 K=1,6
                     DO 37 S=1,8
37                   A6(S)=SPECTA(S)-NRP(S,K,J,I)
                CALL WWIGHT(1,A6,X1)
                X1=X1+MIN+(T-1)*IDB
                CC=0.
                IF(X1.LT.50.)GO TO 101
                IF(X1.GT.76.)X1=75.9
                CC=AITINT(LEV,U,16,X1,2,XT)
101             CONTINUE
C
                DO 26 P=1,3
                SAV2=SAV2+SIMEIN(I,K,M,P,N)*PROBON(I,J,P,N)*CC
26              CONTINUE
39         A2(2)=A2(2)+XDAYS(N)*SAV2
      A2(2)=A2(2)/900.
      PI(2,ICA)=PI(2,ICA)+SEOPLE(I,M)*A2(2)*DISLEV(T)
36    CONTINUE
      RETURN
      END
```

Subroutine SLEEP.

This subroutine is called by RUN and calculates the impacted populations for the sleep interference weighting functions.

```
      SUBROUTINE SLEEP(ICA,IK)
C
       COMMON /DERIV / PEOPLE(3,6),SEOPLE(3,6),POPTOT,
     1                  SAV1,PROBON(3,6,4,3),SIMEIN(3,6,6,4,3),
     1                  SPECTA(8)
       COMMON /OUTPUT/ PI(4,12),IRAN,DIR(18,20)
     1         /INPUTS/TITLE(18),OPTYP(6),PERC(3),
     1         ONTIM(3,3),TIM(3,3),TP9(4,3,3),
     1         RMP9(6,3,3),SPECTR(8),IDB,MIN,MAX,
     1         OPDIFF(3),DISLEV(20),IT,OPTOT
       COMMON /FUNCOM/ TIMEIN(3,6,6,4,3),PEOPLM(3,6),
     1         NRP(8,6,6,3),     NRS(8,6,6,3)
      COMMON /CONST/XDAYS(3),TPER(4),HSWT(3),NCANUM(12)
      INTEGER P,S,T
       REAL NRP,NRS,XLEQ(8,3,6,4),A1(6),A6(8)
      IF(IK.NE.2.AND.IK.NE.3)RETURN
      DO 37 I=1,3
           DO 42 J=1,6
                DO 43 S=1,8
                IF(IK.EQ.2)A6(S)=SPECTA(S)-NRP(S,4,J,I)
   43           IF(IK.EQ.3)A6(S)=SPECTA(S)-NRS(S,4,J,I)
   42      CALL WWIGHT(1,A6,A1(J))
      X3=0.
           DO 36 T=1,IT
           DO 36 N=1,3
           DO 36 J=1,6
           BB=TIM(I,N)*60.*RMP9(J,I,N)
C
           X2=PROBON(I,J,3,N)*60.*TPER(4)
   38      X2=X2-BB
           IF(X2.GT.0.)X1=BB
           IF(X2.LE.0.)X1=X2+BB
           IF(X1.LE.0.)GO TO 36
           IF(X1.GT.120.)X1=120.
           AA=A1(J)+MIN+(T-1)*IDB
           SEL=AA+10.*ALOG10(X1)
           CALL WE(W,SEL,ICA)
           X3=X3+XDAYS(N)*W*DISLEV(T)
           IF(X2.GT.0.)GO TO 38
   36      CONTINUE
      X4=0.
           DO 39 M=1,6
   39      X4=X4+SEOPLE(I,M)
      AA=1.
      IF(IK.EQ.3)AA=HSWT(I)
   37 PI(IK,ICA)=PI(IK,ICA)+X4*X3*AA
      RETURN
      END
```

Subroutine WE

This subroutine is called from ALLEQ, SLEEP, and SPEECH and provides the weighting functions for the twelve weighting schemes evaluated. The routine is passed X and ICA which are the noise exposure and the weighting functions number respectively, and returns W, the weight.

```
      SUBROUTINE WE(W,X,ICA)
C
C
      IF(ICA.EQ.1) W=10.**((X-70.)/10.)
      IF(ICA.EQ.2.AND.X.GE.70.) W=0.025*(X-70.)**2.
      IF(ICA.EQ.3.AND.X.GE.70.) W=.6*(X-70.)**1.2
      IF(ICA.LT.4.OR.ICA.GT.5) GO TO 34
      IF(ICA.EQ.5) X=X+15.
      AA=3.36E-6*10.**(.103*X)
      BB=.2*10.**(.03*X)
      CC=1.43E-4*10.**(.08*X)
      W=AA/(BB+CC)
34    IF(ICA.EQ.6) W=.05*(X-55.)
      IF(ICA.EQ.7) W=.01*(10.**((X-55.)/10.)-1.)
      IF(ICA.EQ.8) W=(2.**((X-55.)/10.)-1.)/3.
      IF(ICA.EQ.9) W=.0125*(X-55.)*2.**((X-55.)/10.)
      IF(ICA.EQ.11)W=.0135*X-.5
      IF(ICA.EQ.12)W=.0110*X-.495
      IF(ICA.EQ.1.AND.X.GT.70.)W=1.
      IF(ICA.EQ.2.AND.X.LT.70.)W=0.
      IF(ICA.EQ.3.AND.X.LT.70.)W=0.
      IF(ICA.GT.5.AND.ICA.LT.10.AND.X.LT.55.)W=0.
      IF(ICA.GT.10.AND.W.GT.1)W=1.
      IF(W.LT.0.)W=0.
C
      RETURN
      END
```

Subroutine WWIGHT

This subroutine is called from ALLEQ, SLEEP, and SPEECH and generates the one-third octave band data from octave band data. The subroutine then calls WT to W-weight the one-third octave band data.

```
      SUBROUTINE WWIGHT(ICA,SPECTR,Y)
      REAL SPL(24),SPECTR(8)
C
C
      DO 20 I=1,8
      DO 20 J=1,3
      K=3*(I-1)+J
   20 SPL(K)=SPECTR(I)-4.77
      CALL WT(SPL,Y)
      RETURN
      END
```

Subroutine WT

This subroutine A-weights the one-third octave band data and then returns an overall A-weighted level.

```
      SUBROUTINE WT(SPL,A)
C
C
      REAL SPL(24),W(24)
      DATA W/-30.2,-26.2,-22.5,
     1      -19.1,-16.1,-13.4,-10.9,-8.6,-6.6,
     1      -4.8,-3.2,-1.9,-.8,0.,.6,1.,1.2,1.3,
     1      1.2,1.,.5,-.1,-1.1,-2.5/
      A=0.
      DO 20 J=1,24
      A=A+10.**((SPL(J)+W(J))/10.)
   20 CONTINUE
      IF(A.LE.0.)A=1.
      A=10.*ALOG10(A)
      RETURN
      END
```

Function AITINT

This function uses Aitkens integration technique to evaluate the speed unintelligibility function given the sound exposure level. The function is input as discrete points.

```
      FUNCTION AITINT(X,Y,N,XB,K,P)
C
C
      REAL X(N),Y(N),P(K)
      DO 2 J=1,N
      IF(X(J).GT.XB)GO TO 3
      IF(X(J).LT.XB)GO TO 2
1     AITINT=Y(J)
      RETURN
2     CONTINUE
      J=N
3     J=MIN0(MAX0(1,J-(K+1)/2),N-K+1)
      IF(J+K.LE.N.AND.X(J+K)-XB.LT.XB-X(J))J=J+MOD(K,2)
      P(1)=Y(J)
C
C
      DO 4 I=2,K
      P(I)=Y(J+I-1)
          DO 4 L=2,I
4         P(I)=(P(L-1)*(X(J+I-1)-XB)-P(I)*(X(J+L-2)-XB))
     1         /((X(J+I-1)-XB)-(X(J+L-2)-XB))
      AITINT=P(K)
      RETURN
      END
```

Subroutine STORE

This subroutine is called by the RUN subroutine and is used to store inputs and outputs onto a temporary output file for later access by the PRINT subroutine.

```
C
C
C
         SUBROUTINE STORE
       COMMON /OUTPUT/ PI(4,12),IRAN,DIR(18,20),NUNIT
        COMMON/COMSG/MAPPRM(100),MAPHLP(100),MAPERR(100),
     1        LEVEL,SINGLE
     1        /INPUTS/TITLE(18),OPTYP(6),PERC(3),
     1        ONTIM(3,3),TIM(3,3),TP9(4,3,3),
     1        RMP9(6,3,3),SPECTR(8),IDB,MIN,MAX,
     1        OPDIFF(3),DISLEV(20),IT
             IRAN=IRAN+1
         WRITE(30'IRAN)TITLE,OPTYP,PERC,
     1  ONTIM,TIM,TP9,
     1  RMP9,SPECT,IDB,MIN,MAX,
     1  OPDIFF,DISLEV,IT,
     1  PI,POPTOT,SAV1
C
       DO 100 III=1,18
100    DIR(III,IRAN)=TITLE(III)
C
       RETURN
       END
```

Subroutine CHANGE

This subroutine is called by OPTION and allows the user to change values of the product specific and fundamental data sets.

```
C
C
C
          SUBROUTINE CHANGE
C
        CALLED BY OPTION
          ALLOWS THE USER TO CHANGE VARIABLE ARRAYS
          OR INDIVIDUAL VARIABLE ELEMENTS FOR
          BOTH PRODUCT SPECIFIC  VARIABLES AND
          FUNDAMENTAL VARIABLES
C
          LOGICAL SINGLE
          DIMENSION VAR(18),RESPO(18),ARRAY(18),NUMSUB(5)
        COMMON/COMSG/MAPPRM(100),MAPHLP(100),MAPERR(100),
     1         LEVEL,SINGLE
        COMMON/CODNAM/HOUSE(6,3),ROOM(6,6),PERSON(6,6),
     1         DAY(6,3),PERIOD(6,4),BAND(6,8),
     1        CODE(6,10),OPTION(2,7),VNAMES(2,17),
     1         SUBS(9),VSIZES(5,17)
        COMMON/INPUTS/TITLE(18),OPTYP(6),PERC(3),
     1         ONTIM(3,3),TIM(3,3),TP9(4,3,3),
     1         RMP9(6,3,3),SPECTR(8),IDB,MIN,MAX,
     1         OPDIFF(3),DISLEV(20),IT,OPTOT
        COMMON /FUNCOM/ TIMEIN(3,6,6,4,3),PEOPLM(3,6),
     1          NRP(8,6,6,3),     NRS(8,6,6,3)
          REAL NRP,NRS
          DATA AAAAA/'A'/,BLANK/' '/
          SINGLE = .TRUE.
C
C         SPECIFY CHANGE VARIABLE
C
1       CALL MSG('PROM',25)
          LEVEL=50
        CALL ASK(DUMMY,VAR,&1,&9999,0)
C
C       CHECK FOR VARIABLE NAME
C
10             DO 100 III=1,17
               IF(VNAMES(1,III).EQ.VAR(1))GOTO 40
100          CONTINUE
```

```
C
C         *** ERROR ***
C         INVALID VARIABLE NAME
C
        CALL MSG('ERRO',6)
          GO TO 1
C
C         IF VARIABLE TO CHANGE IS AN ARRAY OF NUMBERS
C         ASK IF USER WANTS TO CHANGE ALL VALUES IN
C         THE ARRAY OR JUST A SINGLE ELEMENT
C
40        IF(VSIZES(1,III).EQ.BLANK)GO TO 90
          IF(III .GT. 13) GO TO 43
42        WRITE(6,41)
          LEVEL=51
41        FORMAT(' TYPE ''A'' TO CHANGE ALL NUMBERS IN GROUP',/,
     1           '      ''S'' TO CHANGE A SINGLE NUMBER IN GROUP',/)
        CALL ASK(DUMMY,RESPO,&42,&9999,0)
          IF(RESPO(1).EQ.AAAAA)GO TO 90
C
C         WANT TO CHANGE A SINGLE ELEMENT IN THE
C         VARIABLE ARRAY
C         FIND OUT WHICH ELEMENT
C
43              DO 50 JJJ=1,5
               IF(VSIZES(JJJ,III).EQ.BLANK)GO TO 65
50           CALL SCROPT(NUMSUB(JJJ),VSIZES(JJJ,III),&1,&43)
C
C         PROMPT FOR SINGLE ARRAY ELEMENT
C
65        WRITE(6.66)(VNAMES(IJK,III),IJK=1,2)
66        FORMAT(/,16(1H*),//,' ENTER VALUE   FOR:  ',2A4)
        CALL MSG('PROM',III)
          WRITE(6,77)
77        FORMAT(1X)
C
               DO 75 JJJ=1,5
               IF(VSIZES(JJJ,III).EQ.BLANK)GO TO 85
75            CALL NAMIT(NUMSUB(JJJ),VSIZES(JJJ,III))
```

```
C
C         ENTER VALUE FOR ARRAY ELEMENT
C
85        GO TO (9999,200,300,400,500,600,700,800.9999,
     1           9999.9999,9999.1300,1400,1500.1600,1700),III
        CALL MSG('ERRO',7)
          GO TO 997
200          LEVEL=6
        CALL ASK(OPTYP(NUMSUB(1)),ARRAY,&65,&9999.1)
          IF((OPTYP(NUMSUB(1)).GT.1).OR.(OPTYP(NUMSUB(1)).LT.0))
     1    GO TO 1000
          GO TO 997
300          LEVEL=7
        CALL ASK(PERC(NUMSUB(1)),ARRAY,&65,&9999.1)
          IF((PERC(NUMSUB(1)).GT.1).OR.(PERC(NUMSUB(1)).LT.0))
     1    GO TO 1000
          GO TO 997
400          LEVEL=10
        CALL ASK(ONTIM(NUMSUB(1),NUMSUB(2)),ARRAY,&65,&9999,1)
          GO TO 997
500          LEVEL=13
          CALL ASK(TIM(NUMSUB(1),NUMSUB(2)),ARRAY.&65,&9999,1)
          GO TO 997
600          LEVEL=17
          CALL ASK(TP9(NUMSUB(1),NUMSUB(2),NUMSUB(3)),ARRAY,&65,
     1           &9999,1)
          IF((TP9(NUMSUB(1),NUMSUB(2),NUMSUB(3)).GT.1).OR.
     1    (TP9(NUMSUB(1),NUMSUB(2),NUMSUB(3)).LT.0))
     1    GO TO 1000
          GO TO 997
700          LEVEL=21
          CALL ASK(RMP9(NUMSUB(1),NUMSUB(2),NUMSUB(3)),ARRAY,&65.
     1           &9999,1)
          IF((RMP9(NUMSUB(1),NUMSUB(2),NUMSUB(3)).GT.1).OR.
     1    (RMP9(NUMSUB(1),NUMSUB(2),NUMSUB(3)).LT.0))
     1    GO TO 1000
          GO TO 997
800          LEVEL=22
          CALL ASK(SPECTR(NUMSUB(1)),ARRAY,&65.&9999.1)
          GO TO 997
```

```
1300        LEVEL=27
         CALL ASK(DISLEV(NUMSUB(1)),ARRAY,&65,&9999,1)
         GO TO 997
1400        LEVEL=28
         CALL ASK(TIMEIN(NUMSUB(1),NUMSUB(2),NUMSUB(3),
     1           NUMSUB(4),NUMSUB(5)),ARRAY,&65,&9999,1)
         GO TO 997
1500        LEVEL=29
         CALL ASK(PEOPLM(NUMSUB(1),NUMSUB(2)),ARRAY,&65,&9999,1)
         GO TO 997
1600        LEVEL=30
         CALL ASK(NRP(NUMSUB(1),NUMSUB(2),NUMSUB(3),
     1    NUMSUB(4)),ARRAY,&65,&9999,1)
         GO TO 997
1700        LEVEL=31
         CALL ASK(NRS(NUMSUB(1),NUMSUB(2),NUMSUB(3),
     1    NUMSUB(4)),ARRAY,&65,&9999,1)
         GO TO 997
C
C        WANT TO CHANGE ALL ELEMENTS IN THE VARIABLE
C        ARRAY
C
90       CALL QUEST(III)
997      WRITE(6,998)
998      FORMAT(16(1H*),/)
         GO TO 1
C
C      *** ERROR ***
C      MUST ENTER A FRACTION
C
1000       CALL MSG('ERRO',14)
           GO TO 65
9999      RETURN
          END
```

Subroutine INPUT

This subroutine is called by OPTION and controls the input of all product specific variables in the interactive mode.

```
    SUBROUTINE INPUT
C
C   CALLED BY OPTION
C   CONTROLS THE INPUT OF ALL PRODUCT
C   SPECIFIC VARIABLES
C
    LOGICAL SINGLE
    DIMENSION ARRAY(18)
   COMMON/COMSG/MAPPRM(100),MAPHLP(100),MAPERR(100),
  1        LEVEL,SINGLE
C
    NULL=0
    SINGLE=.FALSE.
C
   CALL QUEST(NULL)
    RETURN
    END
```

Subroutine QUEST

This subroutine is called by INPUT or CHANGE and allows the user to access elements of the data sets. The subroutine calls SCROPT to get subscript values from user, calls NAMIT to identify the meaning of the subscript values, and calls ASK to get a value for only one element in a data array.

```
C
C
C
      SUBROUTINE QUEST(N)
C
C     CALLED BY INPUT AND CHANGE
C     PROMPTS FOR NEW VALUES FOR
C     PRODUCT SPECIFIC VARIABLE
C     ARRAYS
C
      DIMENSION ARRAY(18)
      LOGICAL SINGLE
      INTEGER P,S,T
      COMMON/COMSG/MAPPRM(100),MAPHLP(100),MAPERR(100),
     1      LEVEL,SINGLE
      COMMON/CODNAM/HOUSE(6,3),ROOM(6,6),PERSON(6,6),
     1      DAY(6,3),PERIOD(6,4),BAND(6,8),
     1      CODE(6,10),OPTION(2,7),VNAMES(2,17),
     1             SUBS(9),VSIZES(5,17)
      COMMON/INPUTS/TITLE(18),OPTYP(6),PERC(3),
     1      ONTIM(3,3),TIM(3,3),TP9(4,3,3),
     1      RMP9(6,3,3),SPECTR(8),IDB,MIN,MAX,
     1      OPDIFF(3),DISLEV(20),IT,OPTOT
C
C     START HERE IF THE USER IS ONLY CHANGING
C     ONE VARIABLE
C
      IF(.NOT.SINGLE)GO TO 100
C
      GO TO (100,200,300,400,500,600,700,
     1      800,900,1000,1100,1200,1300),N
C
        CALL MSG('ERRO',10)
C
        RETURN
C
```

```
8        FORMAT(16(1H*),/)
9        FORMAT(1H-)
16       FORMAT(1X,18A4)
11       FORMAT(/,1X,6A4,/,16(1H-))
10       FORMAT(//,' ENTER ',2A4)
12       FORMAT(/,1X,12A4,/,16(1H-),8X,16(1H-),8X)
13       FORMAT(/,1X,18A4,/,3(16(1H-),8X))
15       FORMAT(/,'****************',//,' ENTER ',2A4)
C
100      WRITE(6,15)(VNAMES(III,1),III=1,2)
         LEVEL = 5
         CALL MSG('PROM',1)
         CALL ASK(DUMMY,TITLE,&100,&9000,0)
         IF(SINGLE)RETURN
C
200      IDEX=1
210      WRITE(6,15)(VNAMES(III,2),III=1,2)
         LEVEL = 6
         CALL MSG('PROM',2)
         WRITE(6,11) (CODE(I,6),I=1,6)
              DO 250 M=IDEX,6
              IDEX=M
              WRITE(6,16)(PERSON(JJJ,M),JJJ=1,6)
              CALL ASK(OPTYP(M),ARRAY,&200,&9000,1)
              WRITE(6,9)
              IF((OPTYP(M).GT.1.).OR.(OPTYP(M).LT.0.))
     1        GO TO 260
250           CONTINUE
         IF(SINGLE)RETURN
         GO TO 300
C
C        *** ERROR ***
C        MUST ENTER A FRACTION
C
260      CALL MSG('ERRO',16)
         GO TO 210
C
300      IDEX=1
310      WRITE(6,15)(VNAMES(III,3),III=1,2)
         LEVEL = 7
         CALL MSG('PROM',3)
         WRITE(6,11) (CODE(I,1),I=1,6)
              DO 350 I=IDEX,3
              IDEX=I
              WRITE(6,16)(HOUSE(JJJ,I),JJJ=1,6)
              CALL ASK(PERC(I),ARRAY,&300,&9000,1)
```

```
                WRITE(6,9)
                IF((PERC(I).GT.1.).OR.(PERC(I).LT.0.)) GO TO 360
 350            CONTINUE
          IF(SINGLE)RETURN
          GO TO 400
C
C         *** ERROR ***
C         MUST ENTER A FRACTION
C
 360          CALL MSG('ERRO',18)
           GO TO 310
C
 400      WRITE(6,8)
          WRITE(6,420)
          LEVEL = 8
  420     FORMAT(' TYPE ''2'' IF THE MINUTES/DAY THE CONSUMER',/,
     1    ' PRODUCT IS OPERATED DEPENDS ON BOTH THE HOUSE TYPE AND',/,
     1    ' THE DAY TYPE',//,' TYPE ''1'' OTHERWISE')
C
          CALL ASK(XNUM,ARRAY,&400,&9000,1)
          NUM=XNUM
          GO TO (440,450),NUM
C
C         *** ERROR ***
C         ANSWER MUST BE 1 OR 2
C
          CALL MSG('ERRO',8)
          GO TO 400
C
C         VARIABLE DEPENDS ONLY ON HOUSE TYPE
C
 440      WRITE(6,10)(VNAMES(III,4),III=1,2)
          LEVEL = 9
          CALL MSG('PROM',4)
          WRITE(6,11) (CODE(I,1),I=1,6)
                DO 449 I=1,3
                WRITE(6,16)(HOUSE(KKK,I),KKK=1,6)
                CALL ASK(ONTIM(I,1),ARRAY,&400,&9000.1)
                ONTIM(I,2)= ONTIM(I,1)
                ONTIM(I.3)= ONTIM(I,1)
 449            WRITE(6,9)
           GO TO 458
C
C         VARIABLE DEPENDS ON BOTH THE HOUSE TYPE AND THE
C       . DAY TYPE
C
```

```
450     WRITE(6,10)(VNAMES(III,4),III=1,2)
        LEVEL = 10
        CALL MSG('PROM',4)
        WRITE(6,12) (CODE(I,1),I=1,6),(CODE(I,7),I=1,6)
              DO 459 N=1,3
              DO 459 I=1,3
              WRITE(6,16)(HOUSE(II,I),II=1,6),
     1                  (DAY(II,N),II=1,6)
              CALL ASK(ONTIM(I,N),ARRAY,&400,&9000,1)
459           WRITE(6,9)
458      IF(SINGLE)RETURN
C
C
500      WRITE(6,8)
         WRITE(6,501)
         LEVEL = 11
501      FORMAT(' TYPE ''2'' IF THE LENGTH OF TIME(IN MINUTES)',/,
     1    ' THE CONSUMER PRODUCT IS USED DURING EACH',/,
     1    ' OPERATION DEPENDS ON BOTH THE HOUSE TYPE',/,
     1    ' AND THE DAY TYPE',//,
     1     ' TYPE ''1'' OTHERWISE',//)
C
        CALL ASK(XNUM,ARRAY,&500,&9000,1)
        NUM=XNUM
        GO TO (510,520),NUM
C
C       *** ERROR ***
C       ANSWER MUST BE 1 OR 2
C
        CALL MSG('ERRO',8)
        GO TO 500
C
C       VARIABLE DEPENDS ONLY ON THE HOUSE TYPE
C
510     WRITE(6,10) (VNAMES(III,5),III=1,2)
        LEVEL = 12
        CALL MSG('PROM',5)
        WRITE(6,11) (CODE(I,1),I=1,6)
             DO 519 I=1,3
             WRITE(6,16)(HOUSE(III,I),III=1,6)
             CALL ASK(TIM(I,1),ARRAY,&500,&9000,1)
             TIM(I,2)=TIM(I,1)
             TIM(I,3)=TIM(I,1)
519          WRITE(6,9)
        IF(SINGLE)RETURN
        GO TO 600
```

```
C
C       VARIABLE DEPENDS ON BOTH THE HOUSE TYPE AND
C       THE DAY TYPE
C
 520    WRITE(6,10) (VNAMES(III,5),III=1,2)
        LEVEL = 13
        CALL MSG('PROM',5)
        WRITE(6,12) (CODE(I,1),I=1,6),(CODE(I,7),I=1,6)
             DO 529 N=1,3
             DO 529 I=1,3
             WRITE(6,16)(HOUSE(III,I),III=1,6),
     1                 (DAY(III,N),III=1,6)
             CALL ASK(TIM(I,N),ARRAY,&500,&9000.1)
 529         WRITE(6,9)
        IF(SINGLE)RETURN
C
 600    WRITE(6,8)
        WRITE(6.601)
        LEVEL = 14
 601    FORMAT(' TYPE ''3'' IF THE PROBABILITY OF',/,
     1  ' THE CONSUMER PRODUCT BEING OPERATED',/,
     2  ' IN A GIVEN TIME PERIOD DEPENDS ON',/,
     3  ' THE HOUSE TYPE AND DAY TYPE',//,
     4  ' TYPE ''2'' IF THE PROBABILITY OF ',/,
     5  ' THE CONSUMER PRODUCT BEING OPERATED',/,
     6  ' IN A GIVEN TIME PERIOD DEPENDS ONLY ON ',/,
     7  ' THE HOUSE TYPE',//,
     8  ' TYPE ''1'' OTHERWISE',//)
C
        CALL ASK(XNUM,ARRAY,&600.&9000.1)
          NUM=XNUM
C
          GO TO (610,620,630),NUM
C
C       *** ERROR ***
C       ANSWER MUST BE 1.2 OR 3
C
        CALL MSG('ERRO',9)
          GO TO 600
C
C       VARIABLE DEPENDS ON THE TIME PERIOD ONLY
C
 610      IDEX=1
 611    WRITE(6,10) (VNAMES(III,6),III=1,2)
        LEVEL = 15
         CALL MSG('PROM',6)
```

```
           WRITE(6,11) (CODE(I,8),I=1,6)
              DO 619 P=IDEX,4
              IDEX=P
            WRITE(6,16)(PERIOD(III,P),III=1,6)
            CALL ASK(TP9(P,1,1),ARRAY,&600,&9000,1)
              IF ((TP9(P,1,1) .GT. 1) .OR. (TP9(P,1,1) .LT. 0))
     1       GO TO 616
C FILL IN
              DO 618 KKK=1,3
              DO 618 JJJ=1,3
 618          TP9(P,KKK,JJJ)=TP9(P,1,1)
            WRITE(6,9)
 619        CONTINUE
C
               IF(SINGLE)RETURN
               GO TO 700
C
C        *** ERROR ***
C        MUST ENTER A FRACTION
C
 616        CALL MSG('ERRO',19)
            GO TO 611
C
C         VARIABLE DEPENDS ON THE TIME PERIOD AND
C          THE HOUSE TYPE
C
 620            IDEX=1
 621       WRITE(6,10) (VNAMES(III,6),III=1,2)
           LEVEL = 16
            CALL MSG('PROM',6)
           WRITE(6,12) (CODE(I,8),I=1,6),(CODE(I,1),I=1,6)
                DO 629 P=IDEX,4
          IDEX2=1
                IDEX=P
                DO 629 I=IDEX2,3
                IDEX2=I
             WRITE(6,16)(PERIOD(III,P),III=1,6),
     1                         (HOUSE(III,I),III=1,6)
             CALL ASK(TP9(P,I,1),ARRAY.&600,&9000,1)
             IF ((TP9(P,I,1) .GT. 1) .OR. (TP9(P,I,1) .LT. 0))
     1      GO TO 626
C FILLIN
               DO 628 KKK=1,3
               TP9(P,I,2)=TP9(P,I,1)
 628           TP9(P,I,3)=TP9(P,I,1)
            WRITE(6,9)
```

```
629       CONTINUE
C
            IF(SINGLE)RETURN
            GO TO 700
C
C     *** ERROR ***
C     MUST ENTER A FRACTION
C
 626   CALL MSG('ERRO',19)
       GO TO 621
C
C     VARIABLE DEPENDS ON TIME PERIOD, HOUSE TYPE
C     AND DAY TYPE
C
630      IDEX=1
631   WRITE(6,10) (VNAMES(II,6),II=1,2)
      LEVEL = 17
          CALL MSG('PROM',6)
      WRITE(6,13) (CODE(I,8),I=1,6),(CODE(I,1),I=1,6),
     1              (CODE(I,7),I=1,6)
            DO 639 P=IDEX,4
      IDEX2=1
            IDEX=P
            DO 639 I=IDEX2,3
      IDEX3=1
            IDEX2=I
            DO 639 N=IDEX3,3
            IDEX3=N
          WRITE(6,16)(PERIOD(III,P),III=1,6),
     1                 (HOUSE(III,I),III=1,6),
     1                 (DAY(III,N),III=1,6)
C
          CALL ASK(TP9(P,I,N),ARRAY,&600,&9000,1)
         IF ((TP9(P,I,N) .GT. 1) .OR. (TP9(P,I,N) .LT. 0))
     1  GO TO 636
       WRITE(6,9)
639       CONTINUE
            IF(SINGLE)RETURN
          GO TO 700
C
C     *** ERROR ***
C     MUST ENTER A FRACTION
C
 636   CALL MSG('ERRO',19)
       GO TO 631
C
```

```
C
  700   WRITE(6,8)
             WRITE(6,701)
         LEVEL = 18
 701       FORMAT(' TYPE ''3'' IF THE PROBABILITY OF',/,
      1  ' THE CONSUMER PRODUCT BEING OPERATED ',/,
      2  ' IN A GIVEN ROOM DEPENDS ON THE HOUSE TYPE',/,
      3  ' AND THE DAY TYPE',//,
      4  ' TYPE ''2'' IF THE PROBABILITY OF ',/,
      5  ' THE CONSUMER PRODUCT BEING OPERATED',/,
      6  ' IN A GIVEN ROOM DEPENDS ONLY ON ',/,
      7  ' THE HOUSE TYPE',//,
      8  ' TYPE ''1'' OTHERWISE')
C
       CALL ASK(XNUM,ARRAY,&700,&9000,1)
          NUM=XNUM
         GO TO (710,720,730),NUM
C
C        *** ERROR ***
C        ANSWER MUST BE 1,2 OR 3
C
      CALL MSG('ERRO',9)
        GO TO 700
C
C                 VARIABLE DEPENDS ONLY ON THE ROOM TYPE
C
710                   IDEX=1
711       WRITE(6,10) (VNAMES(II,7),II=1,2)
        LEVEL = 19
      CALL MSG('PROM',7)
         WRITE(6,11) (CODE(I,2),I=1,6)
        DO 719 J=IDEX,6
        IDEX=J
      WRITE(6,16)(ROOM(III,J),III=1,6)
      CALL ASK(RMP9(J,1,1),ARRAY,&700,&9000,1)
           IF ((RMP9(J,1,1) .GT. 1) .OR. (RMP9(J,1,1) .LT. 0))
     1    GO TO 716
        DO 718 I=1,3
        DO 718 N=1,3
718     RMP9(J,I,N)=RMP9(J,1,1)
      WRITE(6,9)
719   CONTINUE
        IF (SINGLE)RETURN
        GO TO 800
C
C      *** ERROR ***
```

```
C       MUST ENTER A FRACTION
C
 716     CALL MSG('ERRO',21)
         GO TO 711
C
C         VARIABLE DEPENDS ON THE ROOM TYPE
C         AND THE HOUSE TYPE
C
720       IDEX=1
721      WRITE(6,10) (VNAMES(II,7),II=1,2)
        LEVEL = 20
      CALL MSG('PROM',7)
         WRITE(6,12) (CODE(I,2),I=1,6),(CODE(I,1),I=1,6)
        DO 729 J=IDEX,6
          IDEX2=1
      IDEX=J
        DO 729 I=IDEX2,3
          IDEX2=I
      WRITE(6,16)(ROOM(III,J),III=1,6),
     1             (HOUSE(III,I),III=1,6)
      CALL ASK(RMP9(J,I,1),ARRAY,&700,&9000,1)
           IF ((RMP9(J,I,1) .GT. 1) .OR. (RMP9(J,I,1) .LT. 0))
     1    GO TO 726
        DO 728 N=1,3
        RMP9(J,I,2)=RMP9(J,I,1)
728     RMP9(J,I,3)=RMP9(J,I,1)
            WRITE(6,9)
729   CONTINUE
         IF(SINGLE)RETURN
         GO TO 800
C
C       *** ERROR ***
C       MUST ENTER A FRACTION
C
 726     CALL MSG('ERRO',21)
         GO TO 721
C
C         VARIABLE DEPENDS ON THE ROOM TYPE, THE HOUSE
C         TYPE AND THE DAY TYPE
C
730      IDEX=1
731      WRITE(6.10) (VNAMES(II,7),II=1,2)
        LEVEL = 21
        CALL MSG('PROM'.7)
         WRITE(6,13) (CODE(I,2),I=1,6),(CODE(I,1).I=1,6),
     1                      (CODE(I,7),I=1,6)
```

```
          DO 739 J=1,6
         IDEX2=1
           IDEX=J
          DO 739 I=1,3
         IDEX3=1
           IDEX2=I
          DO 739 N=1,3
            IDEX3=N
        WRITE(6,16)(ROOM(III,J),III=1,6),
     1                  (HOUSE(III,I),III=1,6),
     2                  (DAY(III,N),III=1,6)
C
         CALL ASK(RMP9(J,I,N),ARRAY,&700,&9000,1)
           IF ((RMP9(J,I,N) .GT. 1) .OR. (RMP9(J,I,N) .LT. 0))
     1    GO TO 736
739        WRITE(6,9)
            IF(SINGLE)RETURN
         GO TO 800
C       *** ERROR ***
C       MUST ENTER A FRACTION
C
 736     CALL MSG('ERRO',21)
         GO TO 731
C
C
800     WRITE(6,15)(VNAMES(III,8),III=1,2)
       LEVEL = 22
        CALL MSG('PROM',8)
      WRITE(6,11) (CODE(I,9),I=1,6)
           DO 850 S=1,8
        WRITE(6,16)(BAND(III,S),III=1,2)
        CALL ASK(SPECTR(S),ARRAY,&800,&9000,1)
850    WRITE(6,9)
          IF(SINGLE)RETURN
C
C
900     WRITE(6,15)(VNAMES(III,9),III=1,2)
       LEVEL = 23
        CALL MSG('PROM',9)
        CALL ASK(XIDB,ARRAY,&900,&9000,1)
          IDB=XIDB
          IF(SINGLE)GO TO 1300
C
C
C
```

```
1200       WRITE(6,15)(VNAMES(III,12),III=1,2)
         LEVEL = 26
           CALL MSG('PROM',12)
           CALL ASK(OPDIFF(1),ARRAY,&1200,&9000,1)
C
             IF(SINGLE)RETURN
C
C
1000      WRITE(6,15)(VNAMES(III,10),III=1,2)
         LEVEL = 24
          CALL MSG('PROM',10)
          CALL ASK(XMIN,ARRAY,&1000,&9000,1)
            MIN=XMIN
            IF(SINGLE)GO TO 1300
C
C
1100      WRITE(6,15)(VNAMES(III,11),III=1,2)
         LEVEL = 25
          CALL MSG('PROM',11)
          CALL ASK(XMAX,ARRAY,&1100,&9000,1)
            MAX=XMAX
C
C
1300      NDIF=MAX-MIN
          IF(NDIF.GE.IDB)GO TO 1301
C
C         *** ERROR ***
C         MAX MUST BE GREATER THAN MIN
C
          CALL MSG('ERRO',15)
          GO TO 1000
C
1301        IT=(MAX-MIN)/(IDB)
            IDEX=1
1302           WRITE(6,15)(VNAMES(III.13).III=1,2)
        LEVEL = 27
          CALL MSG('PROM',13)
      WRITE(6,11)(CODE(I.10),I=1,6)
             DO 1350 T=IDEX,IT
            IDEX=T
        LO=MIN+(T-1)*IDB
            IHI=LO+IDB
         WRITE(6,1250)LO,IHI
1250       FORMAT(1X,I3,' - ',I3,' DB')
          CALL ASK(DISLEV(T),ARRAY,&1301,&9000,1)
```

```
                IF((DISLEV(T).GT. 1).OR.(DISLEV(T).LT.0))GO TO 1351
                 WRITE(6,9)
       1350       CONTINUE
                WRITE(6,8)
       9000     RETURN
       C
       C        *** ERROR ***
       C        MUST ENTER A FRACTION
       C
        1351      CALL MSG('ERRO',23)
                  GO TO 1302
       C
                END
```

Subroutine LIST

This subroutine is called by OPTION and controls the listing of all of the product specific input variables at the user's terminal.

```
        SUBROUTINE LIST
        LOGICAL SINGLE
        DIMENSION ARRAY(18)
        COMMON/COMSG/MAPPRM(100),MAPHLP(100),MAPERR(100),
     1         LEVEL,SINGLE
       COMMON /FUNCOM/ TIMEIN(3,6,6,4,3),PEOPLM(3,6),
     1           NRP(8,6,6,3),     NRS(8,6,6,3)
       REAL NRP,NRS
C
         NULL=0
         SINGLE=.FALSE.
C
         CALL SHOW(NULL)
         RETURN
         END
C
C
```

Subroutine VERIFY

This subroutine is called by OPTION and allows the user to list separate variables from the product specific data set at the terminal.

```
                  SUBROUTINE VERIFY
                LOGICAL SINGLE
                DIMENSION VAR(18),RESPO(18),ARRAY(18),NUMSUB(5)
                COMMON/COMSG/MAPPRM(100),MAPHLP(100),MAPERR(100),
          1          LEVEL,SINGLE
          1         /CODNAM/HOUSE(6,3),ROOM(6,6),PERSON(6,6),
          1          DAY(6,3),PERIOD(6,4),BAND(6,8),
          1          CODE(6,10),OPTION(2,7),VNAMES(2,17)
          1        ,SUBS(9),VSIZES(5,17)
          1          /INPUTS/TITLE(18),OPTYP(6),PERC(3),
          1          ONTIM(3,3),TIM(3.3),TP9(4,3,3),
          1          RMP9(6,3,3),SPECTR(8),IDB,MIN,MAX,
          1          OPDIFF(3),DISLEV(20),IT,OPTOT
               COMMON /FUNCOM/ TIMEIN(3,6,6.4,3),PEOPLM(3,6),
          1            NRP(8,6,6,3),     NRS(8,6,6,3)
               REAL NRP,NRS
      C
                DATA AAAAA/'A'/,BLANK/' '/
      C
                   SINGLE = .TRUE.
      C
      1             LEVEL=50
                   CALL MSG('PROM',21)
                CALL ASK(DUMMY,VAR,&1,&9999,0)
      C
      10           DO 100 III=1,17
                   IF(VNAMES(1,III).EQ.VAR(1))GOTO 40
      100          CONTINUE
      C
                   CALL MSG('ERRO',6)
                   GO TO 1
      C
      40          IF(VSIZES(1,III).EQ.BLANK)GO TO 90
      42          WRITE(6,41)
                   LEVEL=51
      41          FORMAT(' TYPE ''A'' TO LIST ALL NUMBERS FROM GROUP',/,
          1              '      ''S'' TO LIST A SINGLE NUMBER FROM GROUP',/)
      C
```

```
              LEVEL = 51
             CALL ASK(DUMMY,RESPO,&42,&9999,0)
C
             IF(RESPO(1).EQ.AAAAA)GO TO 90
C
43           DO 50 JJJ=1,5
             IF(VSIZES(JJJ,III).EQ.BLANK)GO TO 65
50         CALL SCROPT(NUMSUB(JJJ),VSIZES(JJJ,III),&1,&43)
C
65         WRITE(6,66)(VNAMES(IJK,III),IJK=1,2)
66           FORMAT(/,16(1H*),//,2A4)
               CALL MSG('PROM',III)
            WRITE(6,68)
68          FORMAT(1X)
C
             DO 75 JJJ=1,5
             IF(VSIZES(JJJ,III).EQ.BLANK)GO TO 85
75           CALL NAMIT(NUMSUB(JJJ),VSIZES(JJJ,III))
C
85           GO TO (9999,200,300,400,500,600,700,800,9999,
     1                9999,9999,9999,1300,1400,1500,1600,1700),III
             CALL MSG('ERRO',7)
             GO TO 1
200          WRITE(6.22)OPTYP(NUMSUB(1))
             GO TO 1
300          WRITE(6,22)PERC(NUMSUB(1))
             GO TO 1
400          WRITE(6,22)ONTIM(NUMSUB(1),NUMSUB(2))
              GO TO 1
500           WRITE(6.22)TIM(NUMSUB(1).NUMSUB(2))
              GO TO 1
600        WRITE(6,22)TP9(NUMSUB(1),NUMSUB(2),NUMSUB(3))
              GO TO 1
700        WRITE(6,22)RMP9(NUMSUB(1),NUMSUB(2),NUMSUB(3))
              GO TO 1
800           WRITE(6.22)SPECTR(NUMSUB(1))
              GO TO 1
1300          WRITE(6,22)DISLEV(NUMSUB(1))
              GO TO 1
1400          WRITE(6,22)TIMEIN(NUMSUB(1),NUMSUB(2),NUMSUB(3),
     1                  NUMSUB(4),NUMSUB(5))
              GO TO 1
```

```
 1500      WRITE(6,22)PEOPLM(NUMSUB(1),NUMSUB(2))
           GO TO 1
 1600      WRITE(6,22)NRP(NUMSUB(1),NUMSUB(2),NUMSUB(3),
     1     NUMSUB(4))
           GO TO 1
 1700       WRITE(6,22)NRS(NUMSUB(1),NUMSUB(2),NUMSUB(3),
     1       NUMSUB(4))
 22        FORMAT(/,F11.3,//,16(1H*),/)
            GO TO 1
 C
 C
 90      CALL SHOW(III)
          WRITE(6,998)
 998      FORMAT(/,16(1H*),/)
        GO TO 1
 9999      RETURN
           END
```

Subroutine SHOW

This subroutine is called by LIST and VERIFY to print names, descriptions and values of single variables specified. QUEST handles multiple variable specifications.

```
          SUBROUTINE SHOW(N)
         LOGICAL SINGLE
         INTEGER P,S,T
         COMMON/COMSG/MAPPRM(100),MAPHLP(100),MAPERR(100),
      1        LEVEL,SINGLE
      1       /CODNAM/HOUSE(6,3),ROOM(6,6),PERSON(6,6),
      1        DAY(6,3),PERIOD(6,4),BAND(6,8),
      1        CODE(6,10),OPTION(2,7),VNAMES(2,17)
      1     ,SUBS(9),VSI-ZES(5,17)
      1       /INPUTS/TITLE(18),OPTYP(6),PERC(3),
      1       ONTIM(3,3),TIM(3,3),TP9(4,3,3),
      1       RMP9(6,3,3),SPECTR(8),IDB,MIN,MAX,
      1       OPDIFF(3),DISLEV(20),IT,OPTOT
       COMMON /OUTPUT/ PI(4,12),IRAN,DIR(18,20),NUNIT
       COMMON /FUNCOM/ TIMEIN(3,6,6,4,3),PEOPLM(3,6),
      1         NRP(8,6,6,3),     NRS(8,6,6,3)
       REAL NRP,NRS
C
C
C
        IF(.NOT.SINGLE)GO TO 100
C
         GO TO (100,200,300,400,500,600,700,
      1       800,900,1000,1100,1200,1300,1400,1500,1600,1700),N
C
          CALL MSG('ERRO',10)
C
          RETURN
C
10        FORMAT(/,18(1H*),//,1X,2A4)
11        FORMAT(1X,18A4)
12        FORMAT(1X,6A4,3X,F12.4)
15        FORMAT(11(1H-))
19        FORMAT(1X)
C
```

```
      100     WRITE(NUNIT,10)(VNAMES(III,1),III=1,2)
              CALL MSG('PROM',1)
             WRITE(NUNIT,19)
            WRITE(NUNIT,11)TITLE
              IF(SINGLE)RETURN
      C
      200     WRITE(NUNIT,10)(VNAMES(III,2),III=1,2)
              CALL MSG('PROM',2)
              WRITE(NUNIT,19)
              WRITE(NUNIT,11) (CODE(III,6),III=1,6)
               WRITE(NUNIT,15)
              DO 250 M=1,6
              WRITE(NUNIT,12)(PERSON(JJJ,M),JJJ=1,6),OPTYP(M)
      250     CONTINUE
              IF(SINGLE)RETURN
      C
      300     WRITE(NUNIT,10)(VNAMES(III,3),III=1,2)
              CALL MSG('PROM',3)
              WRITE(NUNIT,19)
              WRITE(NUNIT,11) (CODE(III,1),III=1,6)
              WRITE(NUNIT,15)
              DO 350 I=1,3
              WRITE(NUNIT,12)(HOUSE(JJJ,I),JJJ=1,6),PERC(I)
      350     CONTINUE
              IF(SINGLE)RETURN
      C
      400    WRITE(NUNIT,10)(VNAMES(III,4),III=1,2)
             CALL MSG('PROM',4)
              WRITE(NUNIT,19)
            WRITE(NUNIT,410)
      410     FORMAT(21X,13(1H-),' DAY TYPE ',13(1H-),
           1  /,' HOUSE TYPE',10X,'WEEKDAY',7X,'WEEKDAY',7X,'WEEKEND',/,
           1  '----------',11X,'(SCHOOL)',6X,'(NO SCHOOL)')
                DO 459 I=1,3
                 WRITE(NUNIT,13)(HOUSE(II,I),II=1,4),(ONTIM(I,N),N=1,3)
      13      FORMAT(1X,4A4,4X,3(F7.2,7X))
      C
      459       CONTINUE
                IF(SINGLE)RETURN
      C
      C
```

```
500      WRITE(NUNIT,10) (VNAMES(III,5),III=1,2)
          CALL MSG('PROM',5)
         WRITE(NUNIT,19)
           WRITE(NUNIT,410)
          DO 529 I=1,3
          WRITE(NUNIT,13)(HOUSE(III,I),III=1,4),
     1              (TIM(I,N),N=1,3)
529      CONTINUE
         IF(SINGLE)RETURN
C
600      WRITE(NUNIT,10) (VNAMES(II,6),II=1,2)
              CALL MSG('PROM',6)
        DO 639 P=1,4
        WRITE(NUNIT,611)(PERIOD(III,P),III=1,6)
611     FORMAT(/,'-------------------',//,' FOR TIME PERIOD :',6A4,/)
        WRITE(NUNIT,410)
             DO 639 I=1,3
             WRITE(NUNIT,13)(HOUSE(III,I),III=1,4),
     1                  (TP9(P,I,N),N=1,3)
C
639          CONTINUE
             IF(SINGLE)RETURN
C
C
700        WRITE(NUNIT,10) (VNAMES(II,7),II=1,2)
         CALL MSG('PROM',7)
        DO 739 J=1,6
        WRITE(NUNIT,711)(ROOM(III,J),III=1,6)
711     FORMAT(/,'-------------------',//,' FOR SOURCE ROOM :',6A4,/)
         WRITE(NUNIT,410)
         DO 739 I=1,3
         WRITE(NUNIT,13)(HOUSE(III,I),III=1,4),
     1              (RMP9(J,I,N),N=1,3)
C
 739       CONTINUE
            IF(SINGLE)RETURN
C
C
```

```
800        WRITE(NUNIT,10)(VNAMES(III,8),III=1,2)
           CALL MSG('PROM',8)
         WRITE(NUNIT,19)
           WRITE(NUNIT,811)
811        FORMAT(' OCTAVE BAND',2X,'LEVEL(DECIBELS)')
         WRITE(NUNIT,15)
            DO 850 S=1,8
           WRITE(NUNIT,81)(BAND(III,S),III=1,2),SPECTR(S)
81         FORMAT(1X,2A4,4X,F7.2)
850        CONTINUE
           IF(SINGLE)RETURN
C
C
900        WRITE(NUNIT,10)(VNAMES(III,9),III=1,2)
           CALL MSG('PROM',9)
            WRITE(NUNIT,911)IDB
911         FORMAT(/,I4,' DECIBELS')
           IF(SINGLE)RETURN
C
C
1000       WRITE(NUNIT,10)(VNAMES(III,10),III=1,2)
           CALL MSG('PROM',10)
           WRITE(NUNIT,911)MIN
           IF(SINGLE)RETURN
C
C
1100       WRITE(NUNIT,10)(VNAMES(III,11),III=1,2)
           CALL MSG('PROM',11)
            WRITE(NUNIT,911)MAX
          IF(SINGLE)RETURN
C
C
C
1200        WRITE(NUNIT,10)(VNAMES(III,12),III=1,2)
            CALL MSG('PROM',12)
             WRITE(NUNIT,912)OPDIFF(1)
912           FORMAT(/,1X,F7.3,' DECIBELS')
C
            IF(SINGLE)RETURN
C
C
```

```
1300       CONTINUE
            WRITE(NUNIT,10)(VNAMES(III,13),III=1,2)
            CALL MSG('PROM',13)
             WRITE(NUNIT,19)
        WRITE(NUNIT,11)(CODE(III,10),III=1,6)
             WRITE(NUNIT,15)
            DO 1350 T=1,IT
           LO=MIN+(T-1)*IDB
           IHI=LO+IDB
           WRITE(NUNIT,1250)LO,IHI,DISLEV(T)
1250       FORMAT(1X,I3,' - ',I3,' DB',4X,F7.3)
1350        CONTINUE
        WRITE(NUNIT,16)
16      FORMAT(/,'******************',///)
9000    RETURN
C
1400         WRITE(NUNIT,10)(VNAMES(III,14),III=1,2)
             CALL MSG('PROM',14)
             WRITE(NUNIT,19)
C
             DO 1490 N=1,3
             DO 1490 P=1,4
             DO 1490 J=1,6
C
1490          WRITE(NUNIT,1491)J,P,N,((TIMEIN(I,K,J,P,N),
     1                      I=1,3),K=1,6)
1491          FORMAT(' J=',I1,' P=',I1,' N=',I1,
     1        1X,3(2X,F5.1,'/',F5.1,'/',F5.1),/,
     1       13X,3(2X,F5.1,'/',F5.1,'/',F5.1))
C
                RETURN
C
1500           WRITE(NUNIT,10)(VNAMES(III,15),III=1,2)
               CALL MSG('PROM',15)
               WRITE(NUNIT,19)
C
               WRITE(NUNIT,1592)((PEOPLM(I,M),I=1,3),M=1,6)
1592        FORMAT(' HOUSE ACCROSS,PEOPLE DOWN',/,6(10X,3F9.2,/))
             RETURN
C
1600        WRITE(NUNIT,10)(VNAMES(III,16),III=1,2)
            CALL MSG('PROM',16)
             WRITE(NUNIT,19)
C
```

```
              DO 1690 I=1,3
              DO 1690 J=1,6
              DO 1690 K=1,6
C
1690         WRITE(NUNIT,1691)K,J,I,(NRP(S,K,J,I),S=1,8)
1691         FORMAT(' K=',I1,' J=',I1,' I=',I1,2X,8(1X,F6.2))
C
             RETURN
C
1700         WRITE(NUNIT,10)(VNAMES(III,17),III=1,2)
             CALL MSG('PROM',17)
             WRITE(NUNIT,19)
C
             DO 1790 I=1,3
             DO 1790 J=1,6
             DO 1790 L=1,6
C
1790         WRITE(NUNIT,1791)L,J,I,(NRS(S,L,J,I),S=1,8)
C
             RETURN
C
1791         FORMAT(' L=',I1,' J=',I1,' I=',I1,2X,8(1X,F6.2))
        END
```

Subroutine SCROPT

This subroutine is called by VERIFY and CHANGE and requests the user to input each of the multiple valued array inputs.

```
               SUBROUTINE SCROPT(NUMSUB,SUBLET,*,*)
                 INTEGER T
            LOGICAL SINGLE
            DIMENSION VAR(18),RESPO(18),ARRAY(18),LIMIT(9)
            COMMON/COMSG/MAPPRM(100),MAPHLP(100),MAPERR(100),
     1           LEVEL,SINGLE
     1         /CODNAM/HOUSE(6,3),ROOM(6,6),PERSON(6,6),
     1           DAY(6,3),PERIOD(6,4),BAND(6,8),
     1           CODE(6,10),OPTION(2,7),VNAMES(2,17)
     1      ,SUBS(9),VSIZES(5,17)
     1           /INPUTS/TITLE(18),OPTYP(6),PERC(3),
     1           ONTIM(3,3),TIM(3,3),TP9(4,3,3),
     1           RMP9(6,3,3),SPECTR(8),IDB,MIN,MAX,
     1           OPDIFF(3),DISLEV(20),IT
           COMMON /FUNCOM/ TIMEIN(3,6,6,4,3),PEOPLM(3,6),
     1             NRP(8,6,6,3),     NRS(8,6,6,3)
           REAL NRP,NRS
             DATA LIMIT/3,6,6,6,6,3,4,8,0/
             LIMIT(9)=IT
C
             DO 50 KKK=1,9
             IF(SUBLET.EQ.SUBS(KKK))GO TO 150
 50          CONTINUE
            RETURN
 150         GO TO(100,200,300,400,500,600,700,800,900),KKK
C
              CALL MSG('ERRO',11)
              RETURN
C
 100         WRITE(6,60)(CODE(JFK,1),JFK=1,6),LIMIT(KKK)
 60          FORMAT(' ENTER CODE FOR :',6A4,/,' CHOOSE ONE OF '.I1,/)
C
             DO 110 JKK=1,3
             WRITE(6,61)JKK,(HOUSE(MMM,JKK),MMM=1,6)
 110         CONTINUE
  61          FORMAT(' TYPE ',I2,' FOR ',6A4)
              GO TO 5000
C
```

```
  200  WRITE(6,60)(CODE(JFK,2),JFK=1,6),LIMIT(KKK)
          DO 210 JKK=1,6
        WRITE(6,61)JKK,(ROOM(MMM,JKK),MMM=1,6)
  210     CONTINUE
          GO TO 5000
C
  300     WRITE(6,70)(CODE(JFK,3),JFK=1,4),
     1                (CODE(JMN,4),JMN=1,6) ,LIMIT(KKK)
  70       FORMAT(' ENTER CODE FOR :',4A4,'-',6A4,
     1   /,' CHOOSE ONE OF ',I1,/)
          DO 310 JKK=1,6
          WRITE(6,61)JKK,(ROOM(MMM,JKK),MMM=1,6)
  310     CONTINUE
          GO TO 5000
C
  400      WRITE(6,70)(CODE(JFK,3),JFK=1,4),
     1                (CODE(JMN,5),JMN=1,6)   ,LIMIT(KKK)
           DO 410 JKK=1,6
           WRITE(6,61)JKK,(ROOM(MMM,JKK),MMM=1,6)
  410        CONTINUE
            GO TO 5000
C
  500     WRITE(6,60)(CODE(JFK,6),JFK=1,6),LIMIT(KKK)
          DO 510 JKK=1,6
          WRITE(6,61)JKK,(PERSON(MMM,JKK),MMM=1,6)
  510     CONTINUE
          GO TO 5000
C
  600     WRITE(6,60)(CODE(JFK,7),JFK=1,6),LIMIT(KKK)
          DO 610 JKK=1,3
          WRITE(6,61)JKK,(DAY(MMM,JKK),MMM=1,6)
  610     CONTINUE
          GO TO 5000
C
  700     WRITE(6,60)(CODE(JFK,8),JFK=1,6),LIMIT(KKK)
          DO 710 JKK=1,4
          WRITE(6,61)JKK,(PERIOD(MMM,JKK),MMM=1,6)
  710     CONTINUE
          GO TO 5000
C
```

```
800   WRITE(6,60)(CODE(JFK,9),JFK=1,6),LIMIT(KKK)
      DO 810 JKK=1,8
      WRITE(6,61)JKK,(BAND(MMM,JKK),MMM=1,6)
810    CONTINUE
       GO TO 5000
C
900    WRITE(6,60)(CODE(JFK,10),JFK=1,6),IT
       DO 910 T=1,IT
      LO=MIN+(T-1)*IDB
      IHI=LO+IDB
       WRITE(6,901)T,LO,IHI
901    FORMAT(' TYPE ',I3,' FOR ',I3,' TO ',I3,' DB')
910   CONTINUE
C
5000    WRITE(6,80)
80    FORMAT(1X)
      LEVEL=41
        CALL ASK(XNUM,RESPO,$9002,$9001,1)
         NUMSUB=XNUM
      IF(NUMSUB.LE.LIMIT(KKK).AND.NUMSUB.GT.0)GO TO 9000
      CALL MSG('ERRO',54)
      GO TO 150
9000    RETURN
9001    RETURN 1
9002    RETURN 2
        END
```

Subroutine NAMIT

This subroutine is called by VERIFY and CHANGE to print the parameter type and the parameter value at the user's terminal.

```
             SUBROUTINE NAMIT(NUMSUB,SUBLET)
          LOGICAL SINGLE
          DIMENSION VAR(18),RESPO(18),ARRAY(18).NUMMY(10)
          COMMON/COMSG/MAPPRM(100),MAPHLP(100),MAPERR(100),
     1          LEVEL,SINGLE
     1         /CODNAM/HOUSE(6,3).ROOM(6,6),PERSON(6,6),
     1          DAY(6,3),PERIOD(6,4),BAND(6,8),
     1          CODE(6,10),OPTION(2,7),VNAMES(2,17)
     1    ,SUBS(9),VSIZES(5,17)
     1          /INPUTS/TITLE(18),OPTYP(6),PERC(3),
     1          ONTIM(3,3),TIM(3,3),TP9(4,3,3),
     1          RMP9(6,3,3),SPECTR(8),IDB,MIN,MAX,
     1          OPDIFF(3),DISLEV(20),IT
         COMMON /FUNCOM/ TIMEIN(3,6,6,4,3),PEOPLM(3,6),
     1            NRP(8,6,6,3),     NRS(8,6,6,3)
         REAL NRP,NRS
C
           DO 50 KKK=1,9
           IF(SUBLET.EQ.SUBS(KKK))GO TO 150
50         CONTINUE
          RETURN
150        GO TO (100,200,300,400.500,600,700,800,900).KKK
C
            CALL MSG('ERRO',11)
            RETURN
C
100        WRITE(6,60)(CODE(JFK,1),JFK=1,6),(HOUSE(MMM,NUMSUB),MMM=1,6)
60         FORMAT(1X,6A4,' - ',6A4)
C
            GO TO 5000
C
200       WRITE(6,60)(CODE(JFK,2),JFK=1,6),(ROOM(MMM,NUMSUB),MMM=1,6)
               GO TO 5000
```

```
C
300         WRITE(6,70)  (ROOM(MMM,NUMSUB),MMM=1,6)
70         FORMAT(' RECEIVING ROOM - PRIMARY - ',6A4)
            GO TO 5000
C
400          WRITE(6,71)(ROOM(MMM,NUMSUB),MMM=1,6)
71        FORMAT(' RECEIVING ROOM-SECONDARY - ',6A4)
              GO TO 5000
C
500     WRITE(6,60)(CODE(JFK,6),JFK=1,6),(PERSON(MMM,NUMSUB),MMM=1,6)
            GO TO 5000
C
600     WRITE(6,60)(CODE(JFK,7),JFK=1,6),(DAY(MMM,NUMSUB),MMM=1,6)
            GO TO 5000
C
700     WRITE(6,60)(CODE(JFK,8),JFK=1,6),(PERIOD(MMM,NUMSUB),MMM=1,6)
            GO TO 5000
C
800     WRITE(6,60)(CODE(JFK,9),JFK=1,6),(BAND(MMM,NUMSUB),MMM=1,6)
             GO TO 5000
C
900        LO=MIN+(NUMSUB-1)*IDB
            IHI=LO+IDB
             WRITE(6,64)(CODE(JFK,10),JFK=1,6),LO,IHI
64       FORMAT(1X,6A4,' - ',I2,' TO ',I2,' DECIBELS')
C
5000       CONTINUE
9000          RETURN
              END
```

Subroutine FINI

This subroutine is called by OPTION to ask the user if the session is indeed to be ended. If the answer is yes, the program ends execution, if no the subroutine returns to OPTION.

```
           SUBROUTINE FINI(*)
C
C          CALLED BY OPTION
C          REGULATES PROGRAM TERMINATION
C               RETURN 1 - RETURN TO OPTION LEVEL
C               RETURN   - FOR PROGRAM TERMINATION
C
           DATA YES/'YES '/
           DIMENSION ARRAY(18)
1          WRITE(6,10)
10         FORMAT(/,' ARE YOU FINISHED?')
           CALL ASK(DUMMY,ARRAY,&1,&9000,0)
           IF(ARRAY(1).NE.YES)RETURN
           WRITE(6,25)
25         FORMAT(/,' BYE',/)
9000       RETURN 1
          END
```

APPENDIX A. PROGRAM REFERENCES

1. "Procedure for Assessing Impact on Public Health and Welfare Due to Noise Emitted by Household and Consumer Products," Yaniv et al, National Bureau of Standards, Washington, D.C., Report to EPA January 1979

2. WYLBUR User's Manual

APPENDIX B. GLOSSARY OF VARIABLES

B.1. Product Specific Variables

The product specific variables include all the data input that are dependent on the product. These inputs will change from product to product. This table defines each of the product specific variables, and describes any restrictions for the values of the respective variables.

Input Variable	Definition	Restrictions
TITLE	A title describing the product and operator's requirement.	•The title must not be longer than 72 characters. (Only the first 72 characters TITLE will be read).
OPTYP(M)	The fractional distribution of operators among the given person -types (M).	•OPTYP(M) must be a positive fraction. •The fractional distribution of operators must sum to one over all person types.
PERC(I)	The fraction of each house type (I) owning the consumer product.	•Perc(I) must be a positive fraction.
ONTIM(I,N)	The length of time, in minutes/day the consumer product operates as a function of the house type (I) and day type (N).	•None
TIM(I,N)	The length of time, in minutes, the consumer product is used during each operation, as a function of the house type (I) and day type (N).	•None

B.1. Product Specific Variables (cont.)

Input Variables	Definition	Restrictions
TP9(P,I,N)	The probability of the consumer product being operated as a function of the time period (P), the house type (I), and the day type (N).	•TP9(P,I,N) must be a positive fraction. •TP9(P,I,N) must sum to 1 over all time periods (P). $\sum_{P} TP9(P,I,N)=1$ $P=1$
RMP9(J,I,N)	The probability of the consumer product being operated as a function of the room type (J), the house type (I), and the day type (N).	•RMP9(J,I,N) must be a positive fraction. •RMP9(J,I,N) must sum to 1 over all room types (J). $\sum_{J} RMP9(J,I,N)=1$ $J=1$
SPECTR(S)	The octave band level, relative to the A-weighted level, for the consumer product.	•Assume the spectrum shape is the same for the sound power level and the sound pressure level at the operator's location.
IDB	The stepsize of the decibel range in dB.	•The stepsize must be greater than the difference between the maximum and the minimum of the decibel range.
MIN	The minimum value of the decibel range.	•The minimum value of the decibel range must be greater than the difference between the maximum value and the stepsize.

B.1. Product Specific Variables (cont.)

Input Variables	Definition	Restrictions
MAX	The maximum value of the decibel range.	•The maximum value of the decibel range must be greater than the difference between the stepsize and the minimum value.
OPDIFF(1)	The difference between the sound level at the operator location and the power level in the source room.	•The difference between the sound level at the operator location and the power level in the source room is assumed constant.
DISLEV(T)	The probability of the power level in the source room being in a particular decibel range.	•The probability of the power level in the source room being in a particular decibel range must have a non-negative fraction. •The sum of the probabilities of the power level in the source room being a particular decibel range must sum to 1 over all the decibel ranges. $\sum_{T} \text{DISLEV}(T)=1$

B.2. Fundamental Variables

The fundamental variables are those dealing with parameters which are independent of the product in question. This table defines each of the fundamental variables. These variables may be changed, if better data becomes available, through standard WYLBUR commands to replace the FUND.DAT file on USER63 but anyone making such changes should be aware of the permanence of such an undertaking, and its affects on the validity of future runs.

Variable Name	Definition
TIMEIN(I,K,M,P,N)	The length of time, in minutes, that person type (M) spends in room (K) or dwelling type (I) during time period (P) on day (N).
PEOPLM(,M)	The number of people, in millions, or person type (M) living in dwelling type (I).
NRP(S,K,J,I)	The noise reduction for within octave band (S) in the receiving room (K) and the source room (J) or the primary dwelling for dwelling type (I).
NRS(S,L,J,I)	The noise reduction for octave band (S) between the receiving room (L) in the secondary dwelling and the source room (J) in the primary dwelling for the dwelling type (I).

B.3. Input Parameter Codes

The input variables are indexed by various parameters. The parameters, the subscripts associated with each parameter, and the definitions of each individual parameter subscript are described below.

Parameter Category	Parameter Subscript Variable	Definitions of Individual Subscript Variables
House type	I	I=1 Single family I=2 Townhouse I=3 Multifamily
Source room	J	J=1 Kitchen J=2 Living room/Dining room/Family room J=3 Bathroom J=4 Bedroom J=5 Basement/Utility room/Garage J=6 Outdoors
Receiving room in the primary dwelling	K	K=1 Kitchen K=2 Living room/Dining room/Family room K=3 Bathroom K=4 Bedroom K=5 Basement/Utility room/Garage K=6 Outdoors
Receiving room in the secondary dwelling	L	L=1 Kitchen L=2 Living room/Dining room/Family room L=3 Bathroom L=4 Bedroom L=5 Basement/Utility room/Garage L=6 Outdoors

B.3. Input Parameter Codes (cont.)

Parameter Category	Parameter Subscript Variable	Definitions of Individual Subscript Variables
Person type	M	M=1 Unemployed or work at home adult male M=2 Employed (outside the home) adult male M=3 Unemployed or work at home adult female M=4 Employed (outside the home) adult female M=5 School age child M=6 Preschool child at home
Day type	N	N=1 Weekday with school N=2 Weekday with no school N=3 Weekend and holidays
Time Period	P	P=1 7:00 AM - 9:00 AM P=2 9:00 AM - 2:00 PM P=3 2:00 PM - 10:00 PM P=4 10:00 PM - 7:00 AM
Octave band	S	S=1 63 HZ S=2 125 HZ S=3 250 HZ S=4 500 HZ S=5 1000 HZ S=6 2000 HZ S=7 4000 HZ S=8 8000 HZ
The sound power level of the source	T	T=1 MIN-MIN+IDB[1] T=2 T=IT MAX-IDB-MAX[1]

---

[1] The number of sound power levels will defined by the input values MIN, MAX, and IDB (the maximum, minimum and stepsize of the decibel range). The number of levels is defined to be IT where IT is the difference between the maximum and the minimum of the decibel range divided by the stepsize (IT=MAX-MIN/IDB).

(footnote continued on next page)

footnote cont.

Note that the user can not specify for 20 levels. This means that it must not be greater than 20.

The following example describes how the sound levels are defined:

Let MIN=75, MAX=9B and IDB=5
Then IT=3
The parameter subscript T would be:
T=1 75-80 dB
T=2 80-85 dB
T=3 85-90 dB

APPENDIX C. PROMPT FILES, HELP FILES, ERROR FILES

This section contains the prompt messages which are held in random access file CPNIM.PROMPT and are defined in READU and accessed through MSG.

```
 -1
CONSUMER PRODUCT NAME AND OPERATOR REQUIREMENT
 -2
FRACTION OF OPERATORS FOR PERSON TYPE
 -3
FRACTION OF EACH HOUSE THAT OWN THE CONSUMER PRODUCT
 -4
MINUTES/DAY THE CONSUMER PRODUCT OPERATES
 -5
LENGTH OF TIME (IN MINUTES) THE CONSUMER PRODUCT IS USED
DURING EACH OPERATION
 -6
PROBABILITY OF THE CONSUMER PRODUCT BEING OPERATED IN A
GIVEN TIME PERIOD
 -7
PROBABILITY OF THE CONSUMER PRODUCT BEING OPERATED IN A
GIVEN ROOM
 -8
OCTAVE BAND LEVELS FOR THE CONSUMER PRODUCT
 -9
STEPSIZE OF THE DECIBEL RANGE
-10
MINIMUM OF THE DECIBEL RANGE
-11
MAXIMUM OF THE DECIBEL RANGE
-12
DIFFERENCE BETWEEN THE SOUND LEVEL AT THE OPERATOR LOCATION
AND THE POWER LEVEL IN THE SOURCE ROOM
-13
THE PROBABILITY OF THE POWER LEVEL IN THE SOURCE ROOM BEING
IN A PARTICULAR DECIBEL RANGE
-14
TIMEIN - VARIABLE #14
-15
PEOPLM - VARIABLE #15
```

Appendix C cont.

```
-16
NRP - VARIABLE#16
-17
NRS - VARIABLE #17
-20
WHICH VARIABLE DO YOU WANT TO CHANGE?
-21
WHICH VARIABLE DO YOU WANT TO LOOK AT?
-25
WHICH VARIABLE DO YOU WANT TO CHANGE?
-27
YOU HAVE CHOSEN TO DIRECT THE OUTPUT TO A FILE
CPNIM.OUTPUT IS THE NAME OF THE FILE
IF THAT FILE ALREADY EXISTS, THE OUTPUT YOU HAVE REQUESTED WILL
BE ADDED TO THE END OF THAT FILE.

TYPE 'TSOWYL'  - THIS LETS YOU USE WYLBUR - SYSTEM TYPES 'COMMAND?'
TYPE 'USE CPNIM.OUTPUT'  - THIS ACCESSES THE OUTPUT FILE
TYPE 'L' TO LIST THE FILE ON THE TERMINAL
TYPE 'L OFFLINE' TO LIST OUTPUT AT CENTRAL PRINTER
TO STOP THE PRINTING EARLY(BEFORE IT TYPES THE WHOLE THING),
HIT THE BREAK KEY
AFTER PRINTING IS DONE, SYSTEM WILL TYPE 'COMMAND?'
TO RETURN TO TSO, TYPE 'END CLR'
-28
DO YOU WANT THE OUTPUT TO BE PRINTED AT THIS TERMINAL OR
  ON A FILE?  TYPE 'T' OR 'F'
-38
COMPUTATION COMPLETE
DO YOU WANT TO LOOK AT YOUR ANSWERS NOW? (TYPE YES OR NO)
IF YOU ANSWER NO, USE THE PRINT OPTION TO LOOK AT THE OUTPUT LATER.
-50
                          OPERATORS   PRIMARY     SECONDARY   OPERATORS
                                      DWELLING    DWELLING    + PRIMARY
                          ----------  ----------  ----------  ----------
-51
1 FRACTIONAL EXPOSURE WEIGHTING BASED ON AN LEQ OF 70
-52
2 POPULATION WEIGHTED LOSS OF HEARING BASED ON CHABA CRITERION
-53
3 POPULATION WEIGHTED LOSS OF HEARING AT 4000HZ
```

Appendix C cont.

```
-54
4 GENERAL ADVERSE RESPONSE CHABA OUTDOOR NOISE SOURCE
-55
5 GENERAL ADVERSE RESPONSE CHABA INDOOR NOISE SOURCE
-56
6 GENERAL ADVERSE RESPONSE , FRACTIONAL IMPACT METHOD
-57
7 GENERAL ADVERSE RESPONSE , ALEXANDER ENERGY
-58
8 GENERAL ADVERSE RESPONSE , ALEXANDER LOUDNESS
-59
9 GENERAL ADVERSE RESPONSE , ALEXANDER SYNTHESIS
-60
10 SPEECH INTERFERENCE , FRACTIONAL UNINTELLIGIBILITY BASED ON A-LEVEL
-61
11 SLEEP INTERFERENCE , DISRUPTION POTENTIAL FROM SEL
-62
12 SLEEP INTERFERENCE , AWAKENING POTENTIAL FROM SEL
```

Appendix C cont.

This section contains the help messages which are held in random access file CPNIM.HELP and are defined in READU and accessed through MSG.

```
-1

THIS PROGRAM ASSESSES THE HEALTH AND WELFARE
IMPACT OF CONSUMER PRODUCT NOISE.  THE
ORIGINAL VERSION WAS WRITTEN BY THE
NATIONAL BUREAU OF STANDARDS FOR EPA IN
1978.  THE MODEL WAS UPDATED AND MADE
INTERACTIVE IN NATURE IN 1980.

USE OF THIS PROGRAM DOES NOT REQUIRE
EXTENSIVE PROGRAMMING EXPERIENCE.  THE
MODEL IS DESIGNED TO PROMPT THE USER FOR
ALL INPUT DATA, AND ALLOWS MAXIMUM FLEXABILITY
IN PROGRAM OPERATION THROUGH EASILY UNDERSTOOD
DIRECTIONS.

-2

THE OPERATOR IS PROVIDED ADDITIONAL ASSISTANCE
AT ANY OF THE MODEL PROMPTS BY TYPING
     - HELP
     - CODE
     - SOS
     - OOPS
TYPING 'HELP' WILL PROVIDE A MORE EXPANSIVE EXPLANATION
TYPING 'CODE' WILL PROVIDE A LIST OF THE PARAMETER CODES.
TYPING 'SOS ' WILL RETURN THE OPERATOR ONE LEVEL UP.
TYPING 'OOPS' WILL RETURN THE OPERATOR TO THE BEGINNING
OF THE PROMPT.
```

Appendix C cont.

```
 -3

THE OPERATOR CAN GET A LISTING OF THE
ASSISTANCE CODES BY TYPING '?'.


DOCUMENTATION ON THE USE OF THIS PROGRAM, AND A
BATCH VERSION OF THE SAME PROGRAM CAN BE
FOUND IN THE "USERS MANUAL AND PROGRAM GUIDE"
AVAILABLE THROUGH THE EPA-ONAC OFFIC.
 -4

THE USER CAN SPECIFY THE FOLLOWING OPTIONS
      - CHANGE
      - INPUT
      - LIST
      - PRINT
      - RUN
      - STOP
      - VERIFY

THE OPTIONS ARE DEFINED BELOW
      - CHANGE = CHANGE PRODUCT SPECIFIC AND FUNDAMENTAL
                 VARIABLES
      - INPUT  = SPECIFY PRODUCT SPECIFIC VARIABLES
      - LIST   = LIST ALL THE PRODUCT SPECIFIC VARIABLE VALUES
      - PRINT  = PRINT THE OUTPUTS AND/OR INPUTS TO A FILE
                 OR TO THE TERMINAL
      - RUN    = RUN THE MODEL
      - STOP   = END THE EXECUTION OF THE MODEL
      - VERIFY = VERIFY INDIVIDUAL VARIABLE ELEMENT
                 VALUES OR ARRAYS
 -5

ENTER THE NAME OF THE CONSUMER PRODUCT AND WHETHER
AN OPERATOR IS REQUIRED.
THE TITLE MUST NOT BE MORE THAN 72 CHARACTERS.
```

Appendix C cont.

```
 -6

THIS QUESTION WILL PROMPT FOR A NUMBER AFTER IT
SPECIFIES ONE OF THE SIX PERSON TYPES.
AFTER EACH PERSON TYPE, ENTER THE FRACTION
OF TIME THAT PERSON TYPE IS AN OPERATOR.
     EG: UNEMPLOYED ADULT MALE
         .25
THE SUM OF ALL THE FRACTIONS MUST EQUAL ONE.

 -7

THIS QUESTION WILL PROMPT FOR A NUMBER AFTER IT
SPECIFIES ONE OF THE THREE HOUSE TYPES.
AFTER EACH HOUSE TYPE, ENTER THE FRACTION
OF THAT HOUSE TYPE OWNING THE APPLIANCE.
     EG: SINGLE FAMILY
         .34

 -8

ANSWER EITHER '1' OR '2'.
     ENTER '2' IF THE MINUTES/DAY THE CONSUMER PRODUCT
     OPERATES IS A FUNCTION OF BOTH THE HOUSE TYPE
     AND THE DAY TYPE.
          IF THE USER ENTERS '2' THE PROGRAM WILL
          PROMPT FOR NINE NUMBERS.

     ENTER '1' IF THE MINUTES/DAY THE CONSUMER PRODUCT
     OPERATES IN A GIVEN HOUSE IS INDEPENDENT OF THE
     DAY TYPE.
          IF THE USER ENTERS '1' THE PROGRAM WILL
          PROMPT FOR THREE NUMBERS.  THE PROGRAM WILL
          AUTOMATICALLY SET THE MINUTES/DAY THE CONSUMER
          PRODUCT OPERATES EQUAL FOR ALL DIFFERENT DAY
          TYPES IN A GIVEN HOUSE TYPE.
```

Appendix C cont.

```
    -9

   THIS QUESTION WILL PROMPT FOR A NUMBER AFTER
   IT SPECIFIES ONE OF THE THREE HOUSE TYPES.
   AFTER EACH HOUSE TYPE, ENTER THE
   MINUTES/DAY THE CONSUMER PRODUCT OPERATES
   IN THE GIVEN HOUSE TYPE.
        EG: SINGLE FAMILY
            60
   THE MINUTES/DAY THE CONSUMER PRODUCT
   OPERATES SHOULD BE BASED ON AN ANNUAL
   AVERAGE USAGE.

   -10
   THIS QUESTION WILL PROMPT FOR A NUMBER AFTER
   IT SPECIFIES ONE OF THE THREE HOUSE TYPES
   AND ONE OF THE THREE DAY TYPES.
   AFTER EACH HOUSE AND DAY TYPE, ENTER
   THE MINUTES/DAY THE CONSUMER PRODUCT
   OPERATES IN THE GIVEN HOUSE TYPE ON
   THE GIVEN DAY TYPE.
        EG: SINGLE FAMILY           WEEKDAY(SCHOOL)
            60
   THE MINUTES/DAY THE CONSUMER PRODUCT
   OPERATES SHOULD BE BASED ON AN ANNUAL
   AVERAGE USAGE.
   -11

   ANSWER EITHER '1' OR '2'.
        ENTER '2' IF THE LENGTH OF TIME THE CONSUMER
        PRODUCT IS USED DURING EACH OPERATION IS
        IS A FUNCTION OF BOTH THE HOUSE TYPE AND THE
        DAY TYPE.
             IF THE USER ENTERS '2' THE PROGRAM WILL
             PROMPT FOR NINE NUMBERS.

        ENTER '1' IF THE LENGTH OF TIME THE CONSUMER
        PRODUCT IS USED DURING EACH OPERATION IS
        INDEPENDENT OF THE DAY TYPE.
             IF THE USER ENTERS '1' THE PROGRAM WILL
             PROMPT FOR THREE NUMBERS.  THE PROGRAM
             WILL AUTOMATICALLY SET THE LENGTH OF TIME
             THE CONSUMER PRODUCT IS USED DURING
             EACH OPERATION EQUAL FOR ALL DIFFERENT
             DAY TYPES GIVEN A HOUSE TYPE.
```

Appendix C cont.

```
-12

THIS QUESTION WILL PROMPT FOR A NUMBER AFTER
IT SPECIFIES ONE OF THE THREE HOUSE TYPES.
AFTER EACH HOUSE TYPE, ENTER THE
LENGTH OF TIME, IN MINUTES, THE CONSUMER
PRODUCT IS USED DURING EACH OPERATION IN
THE GIVEN HOUSE TYPE.
     EG: SINGLE FAMILY
         60

-13

THIS QUESTION WILL PROMPT FOR A NUMBER AFTER
IT SPECIFIES ONE OF THE THREE HOUSE TYPES AND
ONE OF THE THREE DAY TYPES.
AFTER EACH HOUSE TYPE AND DAY TYPE, ENTER
THE LENGTH OF TIME, IN MUNUTES, THE CONSUMER
PRODUCT IS USED DURING EACH OPERATION IN
THE GIVEN HOUSE TYPE ON THE GIVEN DAY TYPE.
     EG: SINGLE FAMILY          WEEKDAY(SCHOOL)
         60
```

Appendix C cont.

```
-14

ANSWER EITHER '1', '2', OR '3'.
     ENTER '3' IF THE PROBABILITY OF THE CONSUMER
     PRODUCT BEING OPERATED IN A GIVEN TIME PERIOD
     IS A FUNCTION OF BOTH THE HOUSE TYPE AND THE
     DAY TYPE.
          IF THE USER ENTERS '3' THE PROGRAM WILL
          PROMPT FOR ALL THIRTY-SIX NUMBERS.

     ENTER '2' IF THE PROBABILITY OF THE CONSUMER
     PRODUCT BEING OPERATED IN A GIVEN TIME PERIOD
     IS INDEPENDENT OF THE DAY TYPE.
          IF THE USER ENTERS '2' THE PROGRAM WILL
          PROMPT FOR ONLY TWELVE NUMBERS.  THE
          PROGRAM WILL AUTOMATICALLY SET THE
          PROBABILITY OF THE CONSUMER PRODUCT
          BEING OPERATED EQUAL FOR ALL DIFFERENT
          DAY TYPES IN A GIVEN TIME PERIOD
          AND A GIVEN HOUSE TYPE.

     ENTER '1' IF THE PROBABILITY OF THE CONSUMER
     PRODUCT BEING OPERATED IN A GIVEN TIME PERIOD
     IS INDEPENDENT OF BOTH THE HOUSE TYPE AND THE
     DAY TYPE.
          IF THE USER ENTERS '1' THE PROGRAM WILL
          PROMPT FOR ONLY FOUR NUMBERS.  THE
          PROGRAM WILL AUTOMATICALLY SET THE
          PROBABILITY OF THE CONSUMER PRODUCT
          BEING OPERATED EQUAL FOR ALL DIFFERENT
          DAY TYPES AND HOUSE TYPES IN A GIVEN
          TIME PERIOD.
```

```
-15

THIS QUESTION WILL PROMPT FOR A NUMBER AFTER
IT SPECIFIES ONE OF THE FOUR TIME PERIODS.
AFTER EACH TIME PERIOD, ENTER THE PROBABILITY
OF THE CONSUMER PRODUCT BEING OPERATED IN THE
GIVEN TIME PERIOD.
     EG:  700 - 900 HRS
         .1
THE PROBABILITIES SUMMED OVER THE TIME PERIOD
SHOULD EQUAL ONE.
```

Appendix C cont.

```
-16

THIS QUESTION WILL PROMPT FOR A NUMBER AFTER
IT SPECIFIES ONE OF THE FOUR TIME PERIODS AND
ONE OF THE THREE HOUSE TYPES.
AFTER EACH TIME PERIOD AND HOUSE TYPE, ENTER
THE PROBABILITY OF THE CONSUMER PRODUCT BEING
OPERATED IN THE GIVEN TIME PERIOD AND HOUSE TYPE.
     EG:  700 - 900             SINGLE FAMILY
        .1
THE PROBABILITIES SUMMED OVER THE TIME PERIOD
SHOULD EQUAL ONE.

-17

THIS QUESTION WILL PROMPT FOR A NUMBER AFTER
IT SPECIFIES ONE OF THE FOUR TIME PERIODS, ONE
OF THE THREE HOUSE TYPES, AND ONE OF THE THREE
DAY TYPES.
AFTER THE PROMPT, ENTER THE PROBABILITY
OF THE CONSUMER PRODUCT BEING OPERATED IN
THE GIVEN TIME PERIOD IN THE GIVEN HOUSE TYPE
AND ON THE GIVEN DAY TYPE.
     EG: 700 - 900          SINGLE FAMILY           WEEKDAY(SCHOOL)
        .1
THE PROBABILITIES SUMMED OVER THE TIME PERIOD
SHOULD EQUAL ONE.
```

Appendix C cont.

```
  -18

    ANSWER EITHER '1', '2' OR '3'.
         ENTER '3' IF THE PROBABILITY OF THE CONSUMER
         PRODUCT BEING OPERATED IN A PARTICULAR ROOM
         IS A FUNCTION OF BOTH THE HOUSE TYPE AND
         AND THE DAY TYPE.
              IF THE USER ENTERS '3' THE PROGRAM WILL
              PROMPT FOR ALL FIFTY-FOUR NUMBERS.

         ENTER '2' IF THE PROBABILITY OF THE CONSUMER
         PRODUCT BEING OPERATED IN A PARTICULAR ROOM
         IS INDEPENDENT OF THE DAY TYPE.
              IF THE USER ENTER '2' THE PROGRAM WILL
              PROMPT FOR ONLY 18 NUMBERS.  THE PROGRAM
              WILL AUTOMATICALLY SET THE PROBABILITY OF
              THE CONSUMER PRODUCT BEING OPERATED
              EQUAL FOR ALL DIFFERENT DAY TYPES IN A
              GIVEN ROOM AND A GIVEN HOUSE.

         ENTER '1' IF THE PROBABILITY OF THE CONSUMER
         PRODUCT BEING OPERATED IN A PARTICULAR ROOM
         IS INDEPENDENT OF BOTH THE HOUSE TYPE AND THE
         DAY TYPE.
              IF THE USER ENTERS '1' THE PROGRAM WILL
              PROMPT FOR ONLY SIX NUMBERS.  THE PROGRAM WILL
              AUTOMATICALLY SET THE PROBABILITY OF THE
              CONSUMER PRODUCT BEING OPERATED EQUAL
              FOR ALL DIFFERENT HOUSE TYPES AND DAY TYPES
              GIVEN A ROOM TYPE.

  -19

  THIS QUESTION WILL PROMPT FOR A NUMBER AFTER
  IT SPECIFIES ONE OF THE SIX ROOM TYPES.
  AFTER EACH ROOM TYPE, ENTER THE PROBABILITY
  FO THE CONSUMER PRODUCT BEING OPERATED IN THE
  GIVEN ROOM.
       EG: KITCHEN
           .1
  THE PROBABILITIES SUMMED OVER THE ROOM TYPES
  SHOULD EQUAL ONE.
```

Appendix C cont.

-20

```
THIS QUESTION WILL PROMPT FOR A NUMBER
AFTER IT SPECIFIES ONE OF THE SIX ROOM
TYPES AND ONE OF THE THREE HOUSE TYPES.
AFTER THE PROMPT, ENTER THE PROBABILITY OF
THE CONSUMER PRODUCT BEING OPERATED IN
THE GIVEN ROOM AND THE HOUSE TYPE.
     EG: KITCHEN                SINGLE FAMILY
          .1
THE PROBABILITIES SUMMED OVER THE ROOM TYPES
SHOULD EQUAL ONE.
```

-21

```
THIS QUESTION WILL PROMPT FOR A NUMBER AFTER
IT SPECIFIES ONE OF THE SIX ROOM TYPES,ONE OF
THE THREE HOUSE TYPES, AND ONE OF THE
THREE DAY TYPES.  AFTER THE PROMPT,
ENTER THE PROBABILITY OF THE CONSUMER
PRODUCT BEING OPERATED IN THE GIVEN
ROOM, IN THE GIVEN HOUSE TYPE, AND ON
THE GIVEN DAY TYPE.
     EG: KITCHEN              SINGLE FAMILY                     WEEKDAY(SCHOOL)
          .1
THE PROBABILITIES SUMMED OVER THE ROOM TYPES
SHOULD EQUAL ONE.
```

-22

```
THIS QUESTION WILL PROMPT FOR A NUMBER AFTER
IT SPECIFIES ONE OF THE EIGHT OCTAVE BAND LEVELS.
AFTER THE PROMPT, ENTER THE OCTAVE BAND SOUND POWER
LEVEL SPECTRUM, IN DECIBELS, FOR THE PRODUCT CLASS.
     EG: 63 HZ
          80._
```

-23

```
ENTER THE MAXIMUM OF THE SOURCE SOUND POWER
LEVEL  IN DECIBELS RELATIVE TO ONE PICO WATT.
```

Appendix C cont.

```
     -24

     ENTER THE MINIMUM OF THE SOURCE SOUND POWER
     LEVEL IN DECIBELS RELATIVE TO ONE PICO WATT.

     -25

     ENTER THE STEPSIZE OF THE SOURCE SOUND POWER
     LEVEL IN DECIBELS.

     -26

     THE DISTANCE BETWEEN THE SOUND SOURCE AND THE OPERATOR
     POSITION IS ASSUMED TO BE CONSTANT.

     -27
     THE PROBABILITY OF THE POWER LEVEL BEING A
     PARTICULAR DECIBEL RANGE IS TO BE SPECIFIED.
     THIS PROMPT WILL ASK FOR NUMBERS
     AFTER INDICATING THE RANGE OF LEVELS FOR
     THAT PARTICULAR BIN.  IN GENERAL, A 5 DB RANGE
     IS SELECTED.
          EG: 70-75 DB
              .5
     THE PROBABILITY OF THE POWER LEVEL BEING A
     PARTICULAR DECIBEL RANGE MUST SUM TO ONE
     OVER ALL DECIBEL RANGE LEVELS.

     -28

     ENTER THE LENGTH OF TIME, IN MINUTES,
     PERSON TYPE(I), SPENDS IN ROOM(K) OF
     HOUSE TYPE(M) DURING TIME PERIOD(P)
     ON DAY TYPE(N)

     -29

     ENTER THE TOTAL NUMBER OF PEOPLE OF
     PERSON TYPE(I), IN MILLIONS THAT
     LIVE IN HOUSE TYPE(M)
```

Appendix C cont.

```
-30
ENTER THE NOISE REDUCTION WITHIN THE
PRIMARY DWELLING FOR THE OCTAVE BAND
LEVEL(S) BETWEEN RECEIVING ROOM(K) AND
SOURCE ROOM(J) FOR DWELLING TYPE(I).

-31
ENTER THE NOISE REDUCTION WITHIN THE SECONDARY DWELLING
FOR THE OCTAVE BAND LEVEL(S) BETWEEN THE RECEIVING ROOM(L)
IN THE SECONDARY DWELLING AND THE SOURCE ROOM(J) IN THE
PRIMARY DWELLING FOR DWELLING TYPE(I)

-41
THERE IS NO HELP MESSAGE AVAILABLE FOR THIS PROMPT

-50
ENTER ONE OF THESE 17 VARIABLES

TITLE    OPTYP    PERC     ONTIM    TIM     TP9
RMP9     SPECTR   IDB      MIN      MAX     OPDIFF
DISLEV            TIMEIN   PEOPLM   NRP     NRS

TYPE 'SOS' TO RETURN TO THE OPTION LEVEL

-51
TYPE 'A' TO LIST OUT ALL THE NUMBERS IN THIS VARIABLE
TYPE 'S' TO PICK OUT A SINGLE NUMBER IN THE VARIABLE TO CHANGE
         ANYTHING BESIDES 'A' WILL BE CONSIDERED 'S'

-55
THERE IS NO HELP MESSAGE AVAILABLE FOR THIS PROMPT
```

Appendix C cont.

This section contains the error messages which are held in random access file CPNIM.ERROR and are defined in READU and accessed through MSG.

```
*** ERROR NUMBER    1 ***
THAT'S NOT AN OPTION !

*** ERROR NUMBER    2 ***
THAT'S NOT A NUMBER !

*** ERROR NUMBER    3 ***
NO DIGIT FOUND BEFORE BLANK

*** ERROR NUMBER    4 ***
THERE IS AN UPPER LIMIT OF ONE DECIMAL POINT PER NUMBER

*** ERROR NUMBER    5 ***
VALID NUMBER, FOLLOWED BY ANYTHING OTHER THAN SPACES IS NOT ALLOWED

*** ERROR NUMBER    6 ***
THAT'S NOT ONE OF THE VARIABLES. TYPE 'HELP' FOR A LIST OF THE VARIABLES

*** ERROR NUMBER    7 ***
PROGRAM ERROR - IN SUBROUTINE CHANGE OR SUBROUTINE VERIF

*** ERROR NUMBER    8 ***
TYPE 1 OR 2 ONLY, PLEASE

*** ERROR NUMBER    9 ***
TYPE 1 2 OR 3 ONLY, PLEASE

*** ERROR NUMBER   10 ***
PROGRAM ERROR - IN SUBROUTINE QUEST OR SUBROUTINE SHOW

*** ERROR NUMBER   11 ***
PROGRAM ERROR - INVALID SUBSCRIPT LETTER PASSED TO NAMIT OR SCROPT

*** ERROR NUMBER   12 ***
TIMEIN(I,J,M.P,N) IS LESS THAN OPTIM.
YOU MUST CHANGE TIMEIN(I,J.M,P.N) OR
OPTYP(M).  RIGHT NOW SIMEIN(I,J,M,P,N)
IS LESS THAN -.2.

*** ERROR NUMBER   15 ***
MIN GREATER THAN MAX

*** ERROR NUMBER   16 ***
OPTYP MUST BE A POSITIVE FRACTION
```

Appendix C cont.

```
*** ERROR NUMBER   17 ***
OPTYP MUST SUM TO EITHER 0 OR 1
*** ERROR NUMBER   18 ***
PERC MUST BE A POSITIVE FRACTION

*** ERROR NUMBER   19 ***
TP9 MUST BE A POSITIVE FRACTION

*** ERROR NUMBER   20 ***
TP9 MUST SUM TO ONE OVER ALL TIME PERIODS

*** ERROR NUMBER   21 ***
RMP9 MUST BE A POSITIVE FRACTION

*** ERROR NUMBER   22 ***
RMP9 MUST SUM TO ONE OVER ALL ROOM TYPES

*** ERROR NUMBER   23 ***
DISLEV MUST BE A POSITIVE FRACTION

*** ERROR NUMBER   24 ***
MUST CHANGE THE VARIABLE VALUES BEFORE
THE MODEL CAN RUN.

* FOR THE INTERACTIVE MODE
     SPECIFY CHANGE OPTION
     TO CHANGE VARIABLE VALUES
* FOR THE BATCH MODE
     EDIT CPNIN.INPUT
     TO CHANGE VARIABLE VALUES

*** ERROR NUMBER   25 ***
DISLEV MUST SUM TO ONE OVER ALL DECIBEL
RANGE LEVELS

*** ERROR NUMBER   29 ***
YOU MUST COMPUTE ANSWERS BEFORE YOU PRINT THEM OUT
 USE RUN COMMAND BEFORE USING PRINT COMMAND

*** ERROR NUMBER   30 ***
TO REQUEST A SINGLE RUN - TYPE NUMBER OF RUN FOLLOWED BY THE RETURN KEY
TO REQUEST LOT OF RUNS  - TYPE NUMBER OF EACH RUN, SEPARATED BY COMMAS
                          YOU SHOULD NOT TYPE ANYTHING BESIDES COMMAS
                          BETWEEN THE NUMBERS
```

Appendix C cont.

```
*** ERROR NUMBER   34 ***
AT LEAST ONE OF THE RUNS YOU REQUESTED DOES NOT EXIST
TYPE IN THE NUMBER OF THE RUNS YOU WANT, SEPARATED ONLY BY
COMMAS. IF YOU ONLY WANT ONE RUN PRINTED, TYPE IN THE NUMBER
OF THAT RUN.
FOR EXAMPLE, IF YOU WANTED RUNS 1 THROUGH 5, TYPE '1,2,3,4,5'


*** ERROR NUMBER   54 ***
SUBSCRIPT CODE VALUE OUT OF RANGE FOR THIS SUBSCRIPT
```

APPENDIX D. CODE FILE

This file contains code names and parameter codes in file CODDAT.DAT which are read in with subroutine READ10.

```

       THIS FILE CONTAINS
            - NAMES OF THE PARAMETER CODES
            - THE PARAMETER CODES
            - THE OPTION CODES
            - THE VARIABLE NAMES
       THE FILE WILL BE READ ON UNIT 10

       HOUSE TYPES

SINGLE FAMILY
TOWNHOUSE
MULTIFAMILY

       ROOM TYPES

KITCHEN
LR-DR-FR
BATHROOM
BEDROOM
BASEMENT-UTIL-GAR
OUTDOORS

       PERSON TYPES

UNEMPLOYED ADULT MALE
EMPLOYED   ADULT MALE
UNEMPLOYED ADULT FEMALE
EMPLOYED   ADULT FEMALE
SCHOOL     AGE   CHILD
PRESCHOOL        CHILD

       DAY TYPES

WEEKDAY(SCHOOL)
WEEKDAY(NO SCHOOL)
WEEKEND

```

Appendix D cont.

```
39.               PERIOD TYPES
40.
41.       700 -  900 HRS
42.       900 - 1700 HRS
43.      1700 - 2200 HRS
44.      2200 -  700 HRS
45.
46.               OCTAVE BANDS
47.
48.        63 HZ
49.       125 HZ
50.       250 HZ
51.       500 HZ
52.      1000 HZ
53.      2000 HZ
54.      4000 HZ
55.      8000 HZ
56.
57.               PARAMETER SUBSCRIPT CODES
58.
59.      HOUSE TYPE
60.      SOURCE ROOM
61.      RECEIVING ROOM
62.         PRIMARY
63.         SECONDARY
64.      PERSON TYPE
65.      DAY TYPE
66.      PERIOD TYPE
67.      OCTAVE BAND
68.      DB RANGE
69.
70.               OPTIONS
71.
72.      CHANGE
73.      INPUT
74.      LIST
75.      PRINT
76.      RUN
77.      STOP
78.      VERIFY
79.
```

Appendix D cont.

```
 80.              VARIABLE NAMES
 81.
 82.     TITLE
 83.     OPTYP
 84.     PERC
 85.     ONTIM
 86.     TIM
 87.     TP9
 88.     RMP9
 89.     SPECTR
 90.     IDB
 91.     MIN
 92.     MAX
 93.     OPDIFF
 94.     DISLEV
 95.     TIMEIN
 96.     PEOPLM
 97.     NRP
 98.     NRS
 99.
100.              SUBSCRIPTS              PARAMETER CODES
101.
102.     I                                HOUSE TYPE
103.     J                                SOURCE ROOM
104.     K                                RECEIVING ROOM PRIMARY
105.     L                                RECEIVING ROOM SECONDARY
106.     M                                PERSON TYPE
107.     N                                DAY TYPE
108.     P                                PERIOD TYPE
109.     S                                OCTAVE BAND
110.     T                                DB RANGE
111.   -----                              --
```

Appendix D cont.

```
            112.              SUBSCRIPTS                    VARIABLE NAMES
    113.
    114.                                     TITLE
    115.     M                               OPTYP
    116.     I                               PERC
    117.     IN                              ONTIM
    118.     IN                              TIM
    119.     PIN                             TP9
    120.     JIN                             RMP9
    121.     S                               SPECTR
    122.                                     IDB
    123.                                     MIN
    124.                                     MAX
    125.                                     OPDIFF
    126.     T                               DISLEV
    127.     IKJPN                           TIMEIN
    128.     IM                              PEOPLM
    129.     SKJI                            NRP
    130.     SLJI                            NRS
  COMMAND ?
```

APPENDIX E. FUNDAMENTAL DATA SET

This appendix describes the fundamental data set and the way it is handled in the programs CPNIM Interactive and Batch modes. Four variables are described in this data set. These data have come from the original work by NBS [*1*], and have not been changed. They represent what is probably the best available data on movement of people in dwellings and average noise reduction from room to room. The four variables are:

- TIMEIN - The length of time, in minutes, spent in one of three dwelling types I, in one of six room types K, by one of six person types M, during one of four period types P, on one of three day types N. The values in the data set vary first with I, second with K, third with M, fourth with P, and fifth with N. These variations are shown in the data listing.
- PEOPLM - The number of people, in millions, residing in house type I, of person type M.
- NRP - Noise reduction in Octave Band S, between receiving room K in the primary dwelling and source room J for dwelling type I. The values vary first with S, second with K, third with J, and fourth with I.
- NRS - Noise reduction in Octave Band S, between receiving room L in the secondary dwelling, and source room J for dwelling type I. The values vary first with S, second with L, third with J and fourth with I.

These data are stored in WYLBUR card format on a namelist file FUND.DAT. The namelist is named FUNDAT and is read into both the Interactive and Batch programs by calling the subroutine READ11. The data may be accessed through standard WYLBUR commands and edited or it may be changed through the change subroutine of the interactive mode. The first option will permanently alter the data set and should only be undertaken if new, more precise data should become available. The second only changes the data set for the one interactive session and is useful in accessing the impacts of changes in fundamental data on the impact calculation results, but is not, in general use, to be employed.

```
&FUNDAT
TIMEIN=15.3,15.3,15.3,31.,36.,41.,15.1,15.1,15.1,
       48.,49.,48.,6.,5.,0.,0.,0.,0.,
       10.2,10.2,10.2,5.3,5.3,5.3,10.1,10.1,10.1,
       21.2,21.2,21.2,0.,0.,0.,0.,0.,0.,
       25.3,25.3,25.3,31.,31.,31.,15.1,15.1,15.1,
       48.,48.,48.,0.,0.,0.,0.,0.,0.,
       18.6,21.7,21.7,15.3,18.5,18.5,15.,15.,15.,
       31.2,31.2,31.2,6.3,0.,0.,0.,0.,0.,
       18.6,21.7,21.7,18.4,18.5,18.5,15.,15.,15.,
       31.2,31.2,31.2,0.,0.,0.,0.,0.,0.,
       25.3,25.3,25.3,36.,31.,31.,10.,15.1,15.1,
       48.,48.,48.,0.,0.,0.,0.,0.,0.,
       33.1,30.,30.,143.,167.,173.,30.,23.7,23.7,
       45.1,30.,45.1,57.3,35.,7.2,43.,41.5,25.,
       11.5,11.5,11.5,23.1,27.,34.9,10.,10.,10.,
       13.9,13.9,13.9,7.9,4.,0.,7.9,7.9,4.,
       72.3,32.3,89.3,133.,147.,153.,43.7,43.7,43.7,
       45.1,45.1,45.1,28.7,14.3,7.2,43.,22.,10.,
       31.4,31.4,31.4,38.2,42.2,42.2,15.9,15.9,15.9,
       39.7,39.7,39.7,15.9,7.9,7.9,0.,0.,0.,
       15.4,15.4,15.,35.,10.,30.,5.,5.,10.,
       35.,39.7,39.7,10.6,7.9,0.,40.1,39.1,21.,
       53.3,53.,53.,72.3,61.6,55.6,15.,15.,15.,
       165.,165.,165.,0.,0.,0.,60.,60.,60.,
       15.,20.,30.,97.,109.,109.,16.2,16.2,16.2,
       67.8,67.8,67.8,30.,14.5,5.,15.,14.5,9.,
       33.,33.,33.,79.6,90.,97.8,20.,20.,20.,
       39.,39.,39.,10.4,5.2,0.,10.4,5.2,2.6,
       36.,39.,39.,107.,109.,109.,16.2,16.2,16.2,
       67.8,67.8,67.8,10.,5.,5.,10.,5.,5.,
       64.,64.2,64.2,75.6,79.6,79.6,18.3,18.3,18.3,
       21.,21.,21.,14.5,7.3,7.3,0.,0.,0.,
       30.,35.,35.,102.,86.9,106.,18.3,18.3,18.3,
            90.,110.,110.,10.,0.,0.,30.,30.,0.,
       36.,35.,39.,66.,55.9,66.8,15.,15.,16.2,
       120.,120.,120.,0.,0.,0.,10.,10.,0.,
       6.3,10.9,12.9,55.4,52.9,52.9,12.5,12.5,12.5,
       441.,441.,441.,0.,2.,0.,0.,0.,0.,
       7.1,7.1,7.1,53.4,53.4,53.4,15.3,15.3,15.3,
       417.,417.,471.,0.,0.,0.,0.,0.,0.,
       12.9,12.9,12.9,52.9,52.9,52.9,12.5,12.5,12.5,
       441.,441.,441.,0.,0.,0.,0.,0.,0.,
       22.2,22.2,22.2,35.6,35.6,35.6,17.9,17.9,17.9,
       416.,416.,416.,0.,0.,0.,0.,0.,0.,
       10.,10.,10.,35.6,35.6,35.6,5.,5.,5.,
47.          489.,489.,489.,0.,0.,0.,0.,0.,0.,
48.          0.,0.,0.,0.,0.,0.,5.,5.,5.,
49.          535.,535.,535.,0.,0.,0.,0.,0.,0.,
50.          15.3,15.3,15.3,31.,36.,41.,15.1,15.1,15.1,
51.          48.,43.,48.,6.,5.,0.,0.,0.,0.,
52.          10.2,10.2,10.2,5.3,5.3,5.3,10.1,10.1,10.1,
53.          21.2,21.2,21.2,0.,0.,0.,0.,0.,0.,
54.          25.3,25.3,25.3,31.,31.,31.,15.1,15.1,15.1,
55.          48.,48.,48.,0.,0.,0.,0.,0.,0.,
56.          18.6,21.7,21.7,15.3,18.5,18.5,15.,15.,15.,
57.          31.2,31.2,31.2,6.3,0.,0.,0.,0.,0.,
58.          18.6,21.7,21.7,18.4,18.5,18.5,15.,15.,15.,
59.          31.2,31.2,31.2,0.,0.,0.,0.,0.,0.,
60.          25.3,25.3,25.3,36.,31.,31.,10.,15.1,15.1,
61.          48.,48.,48.,0.,0.,0.,0.,0.,0.,
62.          33.1,30.,30.,143.,167.,173.,30.,23.7,23.7,
63.          45.1,30.,45.1,57.3,35.,7.2,43.,41.5,25.,
64.          11.5,11.5,11.5,23.1,27.,34.9,10.,10.,10.,
65.          13.9,13.9,13.9,7.9,4.,0.,7.9,7.9,4.,
66.          72.3,82.3,89.3,133.,147.,153.,43.7,43.7,43.7,
67.          45.1,45.1,45.1,28.7,14.3,7.2,43.,22.,10.,
68.          31.4,31.4,31.4,38.2,42.2,42.2,15.9,15.9,15.9,
69.          39.7,39.7,39.7,15.9,7.9,7.9,0.,0.,0.,
70.          15.4,15.4,15.,35.,10.,30.,5.,5.,10.,
71.          35.,39.7,39.7,10.6,7.9,0.,40.1,39.1,21.,
72.          53.3,53.,53.,72.3,61.6,55.6,15.,15.,15.,
73.          165.,165.,165.,0.,0.,0.,60.,60.,60.,
74.          15.,20.,30.,97.,109.,109.,16.2,16.2,16.2,
75.          67.8,67.8,67.8,30.,14.5,5.,15.,14.5,9.,
76.          33.,33.,33.,79.6,90.,97.8,20.,20.,20.,
77.          39.,39.,39.,10.4,5.2,0.,10.4,5.2,2.6,
78.          36.,39.,39.,107.,109.,109.,16.2,16.2,16.2,
79.          67.8,67.8,67.8,10.,5.,5.,10.,5.,5.,
80.          64.,64.2,64.2,75.6,79.6,79.6,18.3,18.3,18.3,
81.          21.,21.,21.,14.5,7.3,7.3,0.,0.,0.,
82.          30.,35.,35.,102.,86.9,106.,18.3,18.3,18.3,
83.              90.,110.,110.,10.,0.,0.,30.,30.,0.,
84.          36.,35.,39.,66.,55.9,66.8,15.,15.,16.2,
85.          120.,120.,120.,0.,0.,0.,10.,10.,0.,
86.          6.3,10.9,12.9,55.4,52.9,52.9,12.5,12.5,12.5,
87.          441.,441.,441.,0.,2.,0.,0.,0.,0.,
88.          7.1,7.1,7.1,53.4,53.4,53.4,15.3,15.3,15.3,
89.          417.,417.,471.,0.,0.,0.,0.,0.,0.,
90.          12.9,12.9,12.9,52.9,52.9,52.9,12.5,12.5,12.5,
91.          441.,441.,441.,0.,0.,0.,0.,0.,0.,
92.          22.2,22.2,22.2,35.6,35.6,35.6,17.9,17.9,17.9,
93.          416.,416.,416.,0.,0.,0.,0.,0.,0.,
 94.        10.,10.,10.,35.6,35.6,35.6,5.,5.,5.,
 95.        489.,489.,489.,0.,0.,0.,0.,0.,0.,
 96.        0.,0.,0.,0.,0.,0.,5.,5.,5.,
 97.        535.,535.,535.,0.,0.,0.,0.,0.,0.,
 98.        19.5,15.,19.5,38.,38.,38.,15.,15.,15.,
 99.        46.3,46.3,46.3,0.,4.5,0.,0.,0.,0.,
100.        14.2,14.2,14.2,7.4,7.4,7.4,5.2,5.2,5.2,
101.        64.3,64.3,64.3,0.,0.,0.,0.,0.,0.,
102.        19.5,19.5,19.5,38.,38.,38.,15.,15.,15.,
103.        46.3,46.3,46.3,0.,0.,0.,0.,0.,0.,
104.        15.7,15.7,15.7,7.,7.,7.,7.2,7.2,7.2,
105.        58.,58.,58.,0.,0.,0.,0.,0.,0.,
106.        18.6,21.7,15.1,18.4,18.5,39.1,15.,15.,7.2,
107.        31.2,31.2,58.0,0.,0.,0.,0.,0.,0.,
108.        19.5,19.5,25.3,38.,38.,31.,15.,15.,15.1,
109.        46.3,46.3,48.,0.,0.,0.,0.,0.,0.,
110.        30.,30.,37.1,159.,156.,199.,34.1,34.1,34.1,
111.        45.,30.,30.,70.6,41.2,10.2,42.6,62.4,30.1,
112.        43.2,43.2,43.2,94.5,137.,170.,26.4,26.4,26.4,
113.        38.4,38.4,38.4,43.,21.5,0.,42.9,21.4,10.7,
114.        56.,61.,67.1,141.,156.,162.,34.1,34.1,34.1,
115.        88.4,88.4,88.4,25.2,10.2,10.2,53.2,33.2,15.1,
116.        67.1,67.1,67.1,127.,134.,134.,33.2,33.2,33.2,
117.        50.2,50.2,50.2,16.6,8.3,8.3,26.6,13.3,13.3,
118.        15.4,15.4,35.0,35.,10.,205.,5.,5.,20.0,
119.        35.,39.7,50.2,10.6,7.9,0.,40.1,39.1,60.,
120.        56.,53.,53.,109.,90.2,55.6,15.,15.,15.,
121.        165.,165.,165.,0.,0.,0.,53.2,60.,60.,
122.        15.,20.,25.,87.7,82.7,92.7,15.,15.,15.,
123.        88.2,88.2,88.2,20.,20.,5.,5.,0.,0.,
124.        35.2,35.2,35.2,76.6,89.,94.2,20.4,20.4,20.4,
125.        58.5,58.5,58.5,10.4,5.2,0.,10.4,5.2,5.2,
126.        45.,45.,45.,82.7,82.7,82.7,15.,15,15.,
127.        88.2,88.2,88.2,5.,5.,5.,0.,0.,0.,
128.        51.5,54.3,54.3,70.,72.7,72.7,15.8,15.8,15.8,
129.        25.4,25.4,25.4,11.,5.5,5.5,0.,0.,0.,
130.        30.,35.,35.,102.,86.9,129.,18.3,18.3,15.8,
131.        90.,110.,90.,10.,0.,0.,30.,30.,0.,
132.        35.,35.,35.,55.9,55.9,72.,15.,15.,15.,
133.        120.,120.,120.,0.,0.,0.,10.,10.,0.,
134.        22.,10.,10.,16.,20.,22.,7.,10.,7.,
135.        477.,477.,477.,0.,0.,0.,0.,0.,0.,
136.        12.9,12.9,12.9,85.4,85.4,85.4,19.8,19.8,19.8,
137.        368.,368.,368.,0.,0.,0.,0.,0.,0.,
138.        22.,22.,22.,16.,16.,16.,7.,7.,7.,
139.        477.,477.,477.,0.,0.,0.,0.,0.,0.,
140.        36.4,36.4,36.4,56.8,56.8,56.8,21.4,21.4,21.4,
               357.,357.,357.,0.,0.,0.,0.,0.,0.,
               10.,10.,10.,35.6,35.6,35.,5.,5.,5.,
               489.,489.,470.,0.,0.,0.,0.,0.,0.,
               0.,0.,0.,0.,0.,0.,5.,5.,5.,
               535.,535.,535.,0.,0.,0.,0.,0.,0.,
        FEOPLM=7.4,2.2,5.2,25.9,7.8,18.1,21.4,6.4,15.0,
               15.6,4.7,10.9,24.6,7.3,17.2,10.0,3.0,7.0
        NRP   =2.2,2.2,2.2,2.2,2.2,2.2,2.2,2.2,
               10.4,11.0,11.6,11.6,11.6,11.6,11.6,11.6,
               11.7,14.2,19.4,20.8,21.0,21.0,21.0,21.0,
               14.2,16.8,22.0,23.3,23.5,23.6,23.6,23.6,
               15.2,17.7,22.9,24.2,24.4,24.5,24.5,24.5,
               20.2,22.7,27.9,29.2,29.4,29.5,29.5,29.5,
               6.0,6.6,7.1,7.2,7.2,7.2,7.2,7.2,
               6.7,6.7,6.7,6.7,6.7,6.7,6.7,6.7,
               15.4,17.4,20.3,20.8,20.9,20.9,20.9,20.9,
               18.0,19.9,22.9,23.4,23.4,23.5,23.5,23.5,
               19.7,22.3,27.5,28.8,29.0,29.0,29.0,29.0,
               23.9,25.9,28.8,29.3,29.4,29.4,29.4,29.4,
               10.7,13.3,18.5,19.8,20.0,20.1,20.1,20.1,
               12.2,14.1,17.1,17.6,17.7,17.7,17.7,17.7,
               -.3,-.3,-.3,-.3,-.3,-.3,-.3,-.3,
               10.3,11.9,13.9,14.2,14.3,14.3,14.3,14.3,
               12.7,15.3,20.5,21.8,22.0,22.1,22.1,22.1,
               17.7,20.3,25.5,26.8,27.0,27.1,27.1,27.1,
               10.7,13.3,18.5,19.8,20.0,20.1,20.1,20.1,
               12.2,14.1,17.1,17.6,17.7,17.7,17.7,17.7,
               3.0,9.6,11.6,11.9,12.0,12.0,12.0,12.0,
               4.8,4.8,4.8,4.8,4.8,4.8,4.8,4.8,
               17.9,20.4,25.6,26.9,27.1,27.2,27.2,27.2,
               22.0,24.0,26.9,27.4,27.5,27.5,27.5,27.5,
               10.7,15.3,18.5,19.8,20.0,20.1,20.1,20.1,
               13.0,15.6,20.8,22.1,22.3,22.3,22.3,22.3,
               9.5,12.1,17.3,18.6,18.8,18.8,18.8,18.8,
               12.1,14.6,19.8,21.2,21.4,21.4,21.4,21.4,
               6.7,6.7,6.7,6.7,6.7,6.7,6.7,6.7,
               23.9,25.9,28.8,29.3,29.4,29.4,29.4,29.4,
               10.7,13.3,18.5,19.8,20.0,20.1,20.1,20.1,
               12.2,14.1,17.1,17.6,17.7,17.7,17.7,17.7,
               9.5,12.1,17.3,18.6,18.8,18.8,18.8,18.8,
               11.3,13.2,16.1,16.7,16.7,16.7,16.7,16.7,
               12.2,14.1,17.1,17.6,17.7,17.7,17.7,17.7,
               16.7,16.7,16.7,16.7,16.7,16.7,16.7,16.7,
               2.2,2.2,2.2,2.2,2.2,2.2,2.2,2.2,
               10.4,11.0,11.6,11.6,11.6,11.6,11.6,11.6,
               11.7,14.2,19.4,20.8,21.0,21.0,21.0,21.0,
187.        14.2,16.8,22.0,23.3,23.5,23.6,23.6,23.6,
188.        15.2,17.7,22.9,24.2,24.4,24.5,24.5,24.5,
189.        20.2,22.7,27.9,29.2,29.4,29.5,29.5,29.5,
190.        6.0,6.6,7.1,7.2,7.2,7.2,7.2,7.2,
191.        6.7,6.7,6.7,6.7,6.7,6.7,6.7,6.7,
192.        15.4,17.4,20.3,20.8,20.9,20.9,20.9,20.9,
193.        18.0,19.9,22.9,23.4,23.4,23.5,23.5,23.5,
194.        19.7,22.3,27.5,28.8,29.0,29.0,29.0,29.0,
195.        23.9,25.9,28.8,29.3,29.4,29.4,29.4,29.4,
196.        10.7,13.3,18.5,19.8,20.0,20.1,20.1,20.1,
197.        12.2,14.1,17.1,17.6,17.7,17.7,17.7,17.7,
198.        -.3,-.3,-.3,-.3,-.3,-.3,-.3,-.3,
199.        10.3,11.9,13.9,14.2,14.3,14.3,14.3,14.3,
200.        12.7,15.3,20.5,21.8,22.0,22.1,22.1,22.1,
201.        17.7,20.3,25.5,26.8,27.0,27.1,27.1,27.1,
202.        10.7,13.3,18.5,19.8,20.0,20.1,20.1,20.1,
203.        12.2,14.1,17.1,17.6,17.7,17.7,17.7,17.7,
204.        3.0,9.6,11.6,11.9,12.0,12.0,12.0,12.0,
205.        4.8,4.8,4.8,4.8,4.8,4.8,4.8,4.8,
206.        17.9,20.4,25.6,26.9,27.1,27.2,27.2,27.2,
207.        22.0,24.0,26.9,27.4,27.5,27.5,27.5,27.5,
208.        10.7,15.3,18.5,19.8,20.0,20.1,20.1,20.1,
209.        13.0,15.6,20.8,22.1,22.3,22.3,22.3,22.3,
210.        9.5,12.1,17.3,18.6,18.8,18.8,18.8,18.8,
211.        12.1,14.6,19.8,21.2,21.4,21.4,21.4,21.4,
212.        6.7,6.7,6.7,6.7,6.7,6.7,6.7,6.7,
213.        23.9,25.9,28.8,29.3,29.4,29.4,29.4,29.4,
214.        10.7,13.3,18.5,19.8,20.0,20.1,20.1,20.1,
215.        12.2,14.1,17.1,17.6,17.7,17.7,17.7,17.7,
216.        9.5,12.1,17.3,18.6,18.8,18.8,18.8,18.8,
217.        11.3,13.2,16.1,16.7,16.7,16.7,16.7,16.7,
218.        12.2,14.1,17.1,17.6,17.7,17.7,17.7,17.7,
219.        16.7,16.7,16.7,16.7,16.7,16.7,16.7,16.7,
220.         2.2,2.2,2.2,2.2,2.2,2.2,2.2,2.2,
221.        10.4,11.0,11.6,11.6,11.6,11.6,11.6,11.6,
222.        11.7,14.2,19.4,20.8,21.0,21.0,21.0,21.0,
223.        14.2,16.8,22.0,23.3,23.5,23.6,23.6,23.6,
224.        15.2,17.7,22.9,24.2,24.4,24.5,24.5,24.5,
225.        20.2,22.7,27.9,29.2,29.4,29.5,29.5,29.5,
226.        6.0,6.6,7.1,7.2,7.2,7.2,7.2,7.2,
227.        6.7,6.7,6.7,6.7,6.7,6.7,6.7,6.7,
228.        15.4,17.4,20.3,20.8,20.9,20.9,20.9,20.9,
229.        18.0,19.9,22.9,23.4,23.4,23.5,23.5,23.5,
230.        19.7,22.3,27.5,28.8,29.0,29.0,29.0,29.0,
231.        23.9,25.9,28.8,29.3,29.4,29.4,29.4,29.4,
232.        10.7,13.3,18.5,19.8,20.0,20.1,20.1,20.1,
233.        12.2,14.1,17.1,17.6,17.7,17.7,17.7,17.7,
234.                    -.3,-.3,-.3,-.3,-.3,-.3,-.3,-.3,
235.                    10.3,11.9,13.9,14.2,14.3,14.3,14.3,14.3,
236.                    12.7,15.3,20.5,21.8,22.0,22.1,22.1,22.1,
237.                    17.7,20.3,25.5,26.8,27.0,27.1,27.1,27.1,
238.                    10.7,13.3,18.5,19.8,20.0,20.1,20.1,20.1,
239.                    12.2,14.1,17.1,17.6,17.7,17.7,17.7,17.7,
240.                    3.0,9.6,11.6,11.9,12.0,12.0,12.0,12.0,
241.                    4.8,4.8,4.8,4.8,4.8,4.8,4.8,4.8,
242.                    17.9,20.4,25.6,26.9,27.1,27.2,27.2,27.2,
243.                    22.0,24.0,26.9,27.4,27.5,27.5,27.5,27.5,
244.                    10.7,15.3,18.5,19.8,20.0,20.1,20.1,20.1,
245.                    13.0,15.6,20.8,22.1,22.3,22.3,22.3,22.3,
246.                    9.5,12.1,17.3,18.6,18.8,18.8,18.8,18.8,
247.                    12.1,14.6,19.8,21.2,21.4,21.4,21.4,21.4,
248.                    6.7,6.7,6.7,6.7,6.7,6.7,6.7,6.7,
249.                    23.9,25.9,28.8,29.3,29.4,29.4,29.4,29.4,
250.                    10.7,13.3,18.5,19.8,20.0,20.1,20.1,20.1,
251.                    12.2,14.1,17.1,17.6,17.7,17.7,17.7,17.7,
252.                    9.5,12.1,17.3,18.6,18.8,18.8,18.8,18.8,
253.                    11.3,13.2,16.1,16.7,16.7,16.7,16.7,16.7,
254.                    12.2,14.1,17.1,17.6,17.7,17.7,17.7,17.7,
255.                    16.7,16.7,16.7,16.7,16.7,16.7,16.7,16.7,
256.          NRS      =33.4,36.4,45.4,52.4,55.4,56.4,56.4,56.4,
257.                    35.7,38.7,47.7,54.7,57.7,58.7,58.7,58.7,
258.                    32.2,35.2,44.2,51.2,54.2,55.2,55.2,55.2,
259.                    34.7,37.7,46.7,53.7,56.7,57.7,57.7,57.7,
260.                    35.7,38.7,47.7,54.7,57.7,58.7,58.7,58.7,
261.                    20.7,23.7,32.7,39.7,42.7,43.7,43.7,43.7,
262.                    37.9,40.9,49.9,56.9,59.9,60.9,60.9,60.9,
263.                    40.2,43.2,52.2,59.2,62.2,63.2,63.2,63.2,
264.                    36.7,39.7,48.7,55.7,58.7,59.7,59.7,59.7,
265.                    39.3,42.3,51.3,58.3,61.3,62.3,62.3,62.3,
266.                    40.2,43.2,52.2,59.2,62.2,63.2,63.2,63.2,
267.                    25.2,28.2,37.2,44.2,47.2,48.2,48.2,48.2,
268.                    30.9,33.9,42.9,49.9,52.9,53.9,53.9,53.9,
269.                    33.2,36.2,45.2,52.2,55.2,56.2,56.2,56.2,
270.                    29.7,32.7,41.7,43.7,51.7,52.7,52.7,52.7,
271.                    32.3,35.3,44.3,51.3,54.3,55.3,55.3,55.3,
272.                    33.2,36.2,45.2,52.2,55.2,56.2,56.2,56.2,
273.                    18.2,21.2,30.2,37.2,40.2,41.2,41.2,41.2,
274.                    36.1,39.1,48.1,55.1,58.1,59.1,59.1,59.1,
275.                    38.3,41.3,50.3,57.3,60.3,61.3,61.3,61.3,
276.                    34.8,37.8,46.8,53.8,56.8,57.8,57.8,57.8,
277.                    37.4,40.4,49.4,56.4,59.4,60.4,60.4,60.4,
278.                    38.3,41.3,50.3,57.3,60.3,61.3,61.3,61.3,
279.                    38.3,41.3,59.3,57.3,60.3,61.3,61.3,61.3,
280.          23.2,26.3,35.3,42.3,45.3,46.3,46.3,46.3,
281.          37.9,40.9,49.9,56.9,59.9,60.9,60.9,60.9,
282.          40.2,43.2,52.2,59.2,62.2,63.2,63.2,63.2,
283.          40.2,43.2,52.2,59.2,62.2,63.2,63.2,63.2,
284.          36,7,39.7,48.7,55.7,58.7,59.7,59.7,59.7,
285.          39.3,42.3,51.3,58.3,61.3,62.3,62.3,62.3,
286.          40.2,43.2,52.2,59.2,62.2,63,2,63.2,63.2,
287.          25.2,28.2,37.2,44.2,47.2,48.2,48.2,48.2,
288.          27.9,30.9,39.9,46.9,49.9,50.9,50.9,50.9,
289.          30.2,33.2,42.2,49.2,52.2,53.2,53.2,53.2,
290.          26.7,29.7,38.7,45.7,48.7,49.7,49.7,49.7,
291.          29.3,32.3,41.3,48.3,51.3,52.3,52.3,52.3,
292.          30.2,33.2,42.2,49.2,52.2,53.2,53.2,53.2,
293.          24.2,24.2,24.2,24.2,27.2,28.2,28.2,28.2,
294.          23.4,26.4,36.4,42.4,43.4,46.4,46.4,46.4,
295.          25.7,28.7,37.7,44.7,47.7,48.7,48.7,48.7,
296.          22.2,25.2,34.2,34.2,41.2,44.2,45.2,45.2,
297.          24.7,27.7,36.7,43.7,46.7,47.7,47.7,47.7,
298.          25.7,28.7,37.7,44.7,47.7,48.7,48.7,48.7,
299.          20.7,23.7,32.7,39.7,42.7,43.7,43.7,43.7,
300.          27.9,30.9,39.9,46.9,49.9,50.9,50.9,50.9,
301.          30.2,33.2,42.2,49.2,52.2,53.2,53.2,53.2,
302.          26.7,29.7,38.7,45.7,48.7,49.7,49.7,49.7,
303.          29.3,32.3,41.3,48.3,51.3,52.3,52.3,52.3,
304.          30.2,33.2,42.2,49.2,52.2,53.2,53.2,53.2,
305.          25.2,28.2,37.2,44.2,47.2,48.2,48.2,48.2,
306.          20.9,23.9,32.9,39.9,42.9,43.9,43.9,43.9,
307.          23.2,26.2,35.2,42.2,45.2,46.2,46.2,46.3,
308.          19.7,22.7,31.7,38.7,41.7,42.7,42.7,42.7,
309.          22.3,25.3,34.3,41.3,44.3,45.3,45.3,45.3,
310.          23.2,26.2,35.2,42.2,45.2,45.2,46.2,46.2,
311.          18.2,21.2,30.2,37.2,40.2,41.2,41.2,41.2,
312.          26.1,29.1,38.1,45.1,48.1,49.1,49.1,49.1,
313.          28.3,31.3,40.3,47.3,50.3,51.3,51.3,51.3,
314.          24.8,27.8,36.8,43.8,46.8,47.8,47.8,47.8,
315.          27.4,30.4,39.4,46.4,49.4,50.4,50.4,50.4,
316.          28.3,31.3,40.3,47.3,50.3,51.3,51.3,51.3,
317.          23.3,26.3,35.3,42.3,45.3,46.3,46.3,46.3,
318.          27.9,30.9,39.9,46.9,50.9,50.9,50.9,50.9,
319.          30.2,33.2,42.2,49.2,52.2,53.2,53.2,53.2,
320.          26.7,29.7,35.7,45.7,48.7,49.7,49.7,49.7,
321.          29.3,32.3,41.3,48.3,51.3,52.3,52.3,52.3,
322.          30.2,33.2,42.2,49.2,52.2,53.2,53.2,53.2,
323.          25.2,28.2,37.2,44.2,47.2,48.2,48.2,48.2,
324.          27.9,30.9,39.9,46.9,49.9,50.9,50.9,50.9,
325.          30.2,33.2,42.2,49.2,52.2,53.2,53.2,53.2,
                    26.7,29.7,38.7,45.7,48.7,49.7,49.7,49.7,
                    29.3,32.3,41.3,48.3,51.3,52.3,52.3,52.3,
                    30.2,33.2,42.2,49.2,52.2,53.2,53.2,53.2,
                    24.2,24.2,24.2,24.2,27.2,28.2,28.2,28.2
        &END
```

APPENDIX F. SAMPLE OPERATIONAL PROCEDURES

The purpose of this appendix is to provide illustrative information that will give the user a better idea of what may be expected when an attempt is made to run the program. Such matters as program initiation, prompts, and error messages will be covered.

The following conventions will be used:

- Lower case alphanumeric characters underlined indicate a message from the computer
- Upper case alphanumeric characters indicate a response by the user
- Footnotes, indicated by superscript numerals, will be used to explain the points illustrated in the example.

The example now follows.

```
ready
EXEC GO
This program assesses the health and welfare
impact of consumer product noise.  The
original version was written by the National
Bureau of Standards for EPA in 1978.  The
model was updated and made interactive in
nature in 1980.

Use of this program does not require
extensive programming experience.  The
model is designed to prompt the user for
all input data, and allows maximum flexability
in program operation through easily understood
directions.

The operator is provided additional assistance
at any of the model prompts by typing
     - HELP
     - CODE
     - SOS
     - OOPS
Typing 'HELP' will provide a more expansive explanation
Typing 'CODE' will provide a list of the parameter codes.
Typing 'SOS' will return the operator one level up.
Typing 'OOPS' will return the operator to the beginning of
the prompt.
```

Appendix F cont.

```
The operator can get a listing of the
assistance codes by typing '?'.

Documentation on the use of this program, and a
batch version of the same program can be found in
the "users manual and program guide" available
through the EPA-ONAC office.

The options are:

     CHANGE
     INPUT
     LIST
     PRINT
     RUN
     STOP
     VERIFY

Enter Option:

INPUT

****************

enter title

consumer product name and operator requirement

TRIAL RUN

****************

enter optyp
fraction of operators for person type

person type
-----------

unemployed adult male

RETURN[1]

*** error number 3 ***

no digit found before blank[2]

****************

enter optyp
fraction of operators for person type
```

---

[1] The carriage return key was pressed at this point to determine whether default values could be entered by this means.

[2] Obviously, this technique did not work.

```
person type
-----------

unemployed adult male
23

*** error number 16 ***
optyp must be a positive fraction
****************

enter optyp
fraction of operators for person type

person type
-----------

unemployed adult male
2/3

*** error number 2 ***
that's not a number![3]
****************

enter optyp
fraction of operators for person type

person type
-----------

unemployed adult male

.1

employed adult male

.0

unemployed adult female

.8

employed adult female

.2

school age child

0

preschool child

0
****************

enter perc
fraction of each house that own the consumer product
```

---

[3] Note that a fracton was asked for and a fraction was entered, but a fraction does not satisfy the requirements of the program.

```
house type
----------

single family
.9

townhouse
.9

multifamily
1

*****************

type '2' if the minutes/day the consumer
product is operated depends on both the house type and
the day type

type '1' otherwise
1

enter ontim
minutes/day the consumer product operates

house type
-----------

single family
20

townhouse
10

multifamily
30

*****************

type '2' if the length of time (in minutes)
the consumer product is used during each
operation depends on both the house type
and the day type

type '1' otherwise


1

enter tim
length of time (in minutes) the consumer product is used
during each operation

house type
-----------

single family
20

townhouse
10
```

```
multifamily
30

****************

type '3' if the probability of
the consumer product being operated
in a given time period depends on
the house type and day type

type '2' if the probability of
the consumer product being operated
in a given time period depends only on
the house type

type '1' otherwise



1



enter tp9
probability of the consumer product being operated in a
given time period

period type
------------

700 - 900 hrs
0

900 - 1700 hrs
.4

1700 - 2200 hrs
.1

2200 - 700 hrs
0

type '3' if the probability of
the consumer product being operated
in a given room depends on the house type
and the day type

type '2' if the probability of
the consumer product being operated
in a given room depends only on
the house type


type '1' otherwise
1
```

```
enter rmp9
probability of the consumer product being operated in a
given room

source room
-----------

kitchen
.1

lr-dr-fr
1

bathroom
.0000000000000001

bedroom
1

basement-util-gar
.0000000000000001

outdoors
0

****************

enter spectr
octave band levels for the consumer product

octave band
-----------

63 HZ
SOS[4]

The options are:
        CHANGE
        INPUT
        LIST
        PRINT
        RUN
        STOP
        VERIFY
```

---

[4] This response takes the program out of the input subroutine and into the option subroutine.

```
enter option:
run[5]
*** error number 20***
tp9 must sum to one over all time periods[6]
*** error number 22***
rmp9 must sum to one over all room types[6]
*** error number 24 ***
must change the variable values before
the model can run.

* for the interactive mode
     specify change option
     to change variable values
* for the batch mode
     edit CPNIM.INPUT
     to change variable values

The options are:
     CHANGE
     INPUT
     LIST
     PRINT
     RUN
     STOP
     VERIFY

Enter option:
STOP

Are you finished?
YES

RYE
```

---

[5] This was an attempt to run the program with the values that had been entered.

[6] The reader can check the preceeding pages to find the variables and errors referred to.

APPENDIX G. PROBLEMS IN EXECUTION

Since NCC archives files that have not been used over a three-month period, all of the files necessary to operate this program will be archived. The approach to de-archiving the necessary files that requires the least knowledge of computer operations is described as follows. First, consult an IBM utilities manual or the NCC user's manual on how to de-archive files. Next, de-archive the file "CN.EPAJHT.S2KC.GO.CLIST". Then, use the command "EXEC GO". The program will not run, but a message will appear that gives the name of a file that is not available. De-archive this file, and try "EXEC GO" again. This procedure may be used repeatedly until, finally, all of the necessary files have been de-archived.